



**TUGAS AKHIR - ME184834**

**RANCANG BANGUN SISTEM KENDALI *DYNAMIC POSITIONING SYSTEM*  
(DPS) PADA PEMODELAN *SEMI-SUBMERSIBLE PLATFORM*  
MENGUNAKAN LOGIKA FUZZY**

Khalif Aji Puspito  
NRP 0421154000032

Dosen Pembimbing  
Juniarko Prananda, S.T, M.T.  
Achmad Baidowi, S.T., M.T.

**DEPARTEMEN TEKNIK SISTEM PERKAPALAN  
FAKULTAS TEKNOLOGI KELAUTAN  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA  
2019**



*“Halaman Ini Sengaja Dikosongkan”*





**TUGAS AKHIR - ME184834**

**RANCANG BANGUN SISTEM KENDALI *DYNAMIC POSITIONING SYSTEM (DPS)* PADA PEMODELAN *SEMI-SUBMERSIBLE PLATFORM* MENGGUNAKAN LOGIKA FUZZY**

Khalif Aji Puspito  
NRP. 0421154000032

Dosen Pembimbing  
Juniarko Prananda, S.T., M.T.  
Achmad Baidowi, S.T., M.T.

DEPARTEMEN TEKNIK SISTEM PERKAPALAN  
Fakultas Teknologi Kelautan  
Institut Teknologi Sepuluh Nopember  
Surabaya  
2019

*“Halaman Ini Sengaja Dikosongkan”*



**FINAL PROJECT - ME184834**

**DESIGN OF DYNAMIC POSITIONING SYSTEM'S CONTROL OF  
PLATFORM SEMI-SUBMERSIBLE MODEL USING FUZZY LOGIC  
CONTROL.**

Khalif Aji Puspito  
NRP. 0421154000032

Advisor :  
Juniarko Prananda, S.T, M.T.  
Achmad Baidowi, S.T., M.T.

DEPARTMENT OF MARINE ENGINEERING  
Faculty of Marine Technology  
Institut Teknologi Sepuluh Nopember  
Surabaya  
2019

*“Halaman Ini Sengaja Dikosongkan”*



**LEMBAR PENGESAHAN**

**RANCANG BANGUN SISTEM KENDALI *DYNAMIC POSITIONING SYSTEM (DPS)* PADA PEMODELAN *SEMI-SUBMERSIBLE PLATFORM* MENGGUNAKAN LOGIKA FUZZY**

Diajukan untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Teknik  
Pada  
Bidang Studi Marine Electrical and Automation System (MEAS)  
Program Studi S-1 Departemen Teknik Sistem Perkapalan  
Fakultas Teknologi Kelautan  
Institut Teknologi Sepuluh Nopember

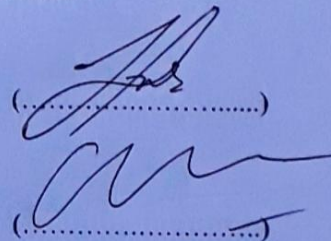
Oleh :

**Khalif Aji Puspito**  
**NRP. 0421154000032**

Disetujui oleh Pembimbing Tugas Akhir :

Juniarko Prananda S.T. M.T.  
NIP. 1990 0605 2015 04 1001

Achmad Baidowi, S.T., M.T.  
NIP.

  
(.....)

SURABAYA  
Juli 2019

*“Halaman Ini Sengaja Dikosongkan”*

## LEMBAR PENGESAHAN

### RANCANG BANGUN SISTEM KENDALI *DYNAMIC POSITIONING SYSTEM (DPS)* PADA PEMODELAN *SEMI-SUBMERSIBLE PLATFORM* MENGGUNAKAN LOGIKA FUZZY

#### SKRIPSI

Diajukan untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Teknik  
Pada  
Bidang Studi Marine Electrical and Automation System (MEAS)  
Program Studi S-1 Departemen Teknik Sistem Perkapalan  
Fakultas Teknologi Kelautan  
Institut Teknologi Sepuluh Nopember

Oleh :

**Khalif Aji Puspito**

**NRP. 0421154000032**

Disetujui oleh :

~~Kepala Departemen Teknik Sistem Perkapalan~~



**Dr. Eng. M. Badrus Zaman, S.T., M.T.**

**NIP. 1977 0802 2008 01 1007**

*“Halaman Ini Sengaja Dikosongkan”*

# **RANCANG BANGUN SISTEM KENDALI *DYNAMIC POSITIONING SYSTEM (DPS)* PADA PEMODELAN *SEMI-SUBMERSIBLE PLATFORM* MENGGUNAKAN LOGIKA FUZZY**

Nama Mahasiswa : Khalif Aji Puspito  
NRP : 0421154000032  
Dosen Pembimbing I : Juniarko Prananda S.T. M.T.  
Dosen Pembimbing II : Achmad Baidowi, S.T., M.T.

## **ABSTRAK**

Rancang bangun sistem kendali *Dynamic Positioning System (DPS)* pada Model *Semi-Submersible Platform* menggunakan *Fuzzy Logic Control* merupakan sistem kendali otomatis untuk mempertahankan posisi dan arah hadap platform. Sistem DPS yang diterapkan pada platform yang beroperasi dikedalaman lebih dari 300 m dapat meningkatkan efektifitas dan nilai ekonomis dibandingkan dengan menggunakan sistem mooring dengan jangkar. Diharapkan platform dengan kontrol Fuzzy mampu merepresentasikan respon pergerakan platform terhadap pergeseran posisi dan arah hadap platform. Untuk mempertahankan posisi dan arah hadap platform, digunakan sensor *GPS* dan *Compass* untuk memberikan feedback pada 4 motor *thruster* dan 4 motor *servo*. Sistem *DGPS (Differential Global Positioning System)* digunakan untuk mengurangi nilai eror sensor GPS dengan menambahkan titik referensi GPS yang terdapat didarat. Pengujian yang dilakukan untuk mengetahui respon platform adalah dengan cara menggeser platform pada empat skenario pergeseran yaitu pergeseran *Surgee*, *Sway*, *Yaw*, serta kombinasi antara *Surge* dan *Sway* pada jarak 2m dari Set Point. Hasil yang didapatkan dari pengujian adalah platform mampu merepresentasikan respon pergerakan dari pergeseran yang dilakukan untuk kembali menuju ke posisi dan sudut heading awal. Nilai eror jarak dari posisi awal dapat mencapai 0.22 m dengan rata-rata nilai eror 0.55m dan rata-rata eror sudut heading adalah 8.33°, serta respon waktu tercepat adalah 23 detik pada rpson pergerakan mundur dengan rata-rata respon waktu pergerakan dari berbagai arah adalah 36 detik.

**Kata Kunci :** *dynamic positioning system, semi-submersible, Fuzzy Logic Control.*

*“Halaman Ini Sengaja Dikosongkan”*

# **DESIGN OF DYNAMIC POSITIONING SYSTEM'S CONTROL OF SEMI-SUBMERSIBLE PLATFORM MODEL USING FUZZY LOGIC CONTROL.**

*Student Name* : Khalif Aji Puspito  
*NRP* : 0421154000032  
*Supervisor I* : Juniarko Prananda S.T. M.T.  
*Supervisor II* : Achmad Baidowi, S.T., M.T.

## **ABSTRACT**

*Design of the Dynamic Positioning System (DPS) control on the Semi-Submersible Platform Model using Fuzzy Logic Control is an automatic control system to maintain the position and heading of the platform. The DPS system that is applied to platforms that operate in depths of more than 300 m can increase the effectiveness and economic value compared to using the mooring system with anchors. It is expected that Fuzzy control platforms are able to represent the response of platform movements to shifting positions and direction to the platform. In order to maintain the position and heading of the platform, GPS sensors and Compass are used to provide feedback on 4 thruster motors and 4 servo motors. The DGPS (Differential Global Positioning System) system was used to reduce the error value of a GPS sensor by adding GPS reference points found on the ground. The test to find out the response of the platform is to shift the platform to four shifting scenarios, namely shifting Surge, Sway, Yaw, and a combination of Surge and Sway at a distance of 2m from Set Point. The results obtained from the test were that the platform is able to represent the movement response from the shift made to return to the position and angle of the initial heading. The value of error distance from the initial position can reach 0.22m with an average error value of 0.55m and the average error of the heading angle is 8.33 °, and the response of the fastest time is 23 seconds in the backward response with the average response time of movement from various directions is 36 seconds.*

**Keywords : dynamic positioning system, semi-submersible, logika fuzzy.**

*“Halaman Ini Sengaja Dikosongkan”*



## KATA PENGANTAR

Alhamdulillah, puji dan syukur kepada Allah SWT karena atas rahmat, taufik dan hidayahNYA, penulis dapat menyelesaikan Tugas Akhir ini dengan judul **“RANCANG BANGUN SISTEM KENDALI *DYNAMIC POSITIONING SYSTEM (DPS)* PADA MODEL *SEMI-SUBMERSIBLE PLATFORM* MENGGUNAKAN LOGIKA FUZZY”** sebagai salah satu syarat dalam menyelesaikan pendidikan sarjana di Departemen Teknik Sistem Perkapalan, Fakultas Teknologi Kelautan, Institut Teknologi, Sepuluh Nopember.

Dalam proses menyelesaikan Tugas Akhir ini, penulis mengucapkan banyak terimakasih kepada :

1. Dwi Atmoko, Prautami Pintaningsih selaku orang tua penulis yang selalu memberikan dukungan Do'a, motivasi, serta materi. Khafid Hidayat dan Hanifah Annisa Mutlasih selaku saudara dan saudari penulis yang selalu mendukung, menemani, dan mendoakan penulis.
2. Bapak Juniarko Prananda, S.T, M.T. dan Bapak Achmad Baidowi, S.T, M.T. selaku dosen pembimbing Tugas Akhir penulis atas bimbingan ilmu dan arahnya kepada penulis.
3. Bapak Dr. Eng Muhammad Badrus Zaman, ST., MT. selaku kepala Departemen Teknik Sistem Perkapalan, FTK, ITS.
4. Bapak Ir. Hari Prastowo, M.Sc. selaku dosen wali yang selalu memberikan arahan dan motivasi selama penulis menempuh pendidikan di departemen ini.
5. Luthfi Halim selaku *partner* penulis yang selalu setia menemani, memotivasi, dan membimbing penulis selama proses penelitian ini sampai selesai.
6. Alfina Shabrin yang selalu setia menemani, memberikan dukungan serta motivasi kepada penulis.
7. Keluarga besar Tim Barunastra ITS, UKM Robotika ITS, Salvage 15, dan *Laboratorium Marine Electrical and Automation System* selaku keluarga terdekat dan tempat penulis mengembangkan diri selama berkuliah di ITS.
8. Dan semua pihak yang telah membantu menyelesaikan tugas akhir ini, yang tidak dapat penulis sebutkan satu persatu.

Penulis menyadari bahwa Tugas Akhir ini masih jauh dari kesempurnaan, sehingga kritik dan saran yang bersifat membangun sangat diharapkan. Akhir kata semoga laporan ini dapat bermanfaat bagi banyak pihak.

Surabaya, 16 Juli 2019

Khalif Aji Puspito

*“Halaman Ini Sengaja Dikosongkan”*

## DAFTAR ISI

LEMBAR PENGESAHAN.....	v
LEMBAR PENGESAHAN.....	vii
ABSTRAK .....	ix
<i>ABSTRACT</i> .....	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR .....	xvii
DAFTAR TABEL .....	xix
BAB I. PENDAHULUAN .....	1
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah .....	2
1.3. Tujuan.....	2
1.4. Batasan Masalah.....	2
BAB II. STUDY LITERATUR.....	3
2.1. <i>Dynamic Positioning System</i> .....	3
2.1.1. <i>Classification Notation</i> .....	4
2.1.2. <i>Position Reference System</i> .....	5
2.2. <i>Semi-Submersible Platform</i> .....	8
2.3. <i>Azimuth Propulsion System</i> .....	9
2.4. <i>Microcontroller STM32 F407</i> .....	9
2.5. <i>Microcontroler Arduino</i> .....	10
2.6. Sensor GPS.....	10
2.7. Sensor Kompas CMPS 11 .....	11
2.8. Motor Servo.....	11
2.9. Motor DC Brushless Watertight.....	12
2.10. <i>ESC (Electronic Speed Controller)</i> .....	13
2.11. Baterai <i>Lithium Polymer</i> .....	13
2.12. Modul <i>Nirkabel HC-12</i> .....	13
2.13. CoCoX IDE (CoIDE).....	14
2.14. <i>Fuzzy Logic Control</i> .....	14
BAB III. METODOLOGI PENELITIAN.....	17
3.1. Metode Penelitian.....	17
3.2. Study Literatur.....	18
3.3. Penentuan Studi Kasus dan Pengumpulan Data .....	18
3.4. Desain System .....	18
3.5. Pembuatan Model dan Program Sistem Kontrol .....	19
3.6. Skenario Uji Coba Model.....	19
3.7. Analisa Hasil Uji Coba dan Perhitungan Data .....	21
3.8. Validasi.....	21
BAB IV RANCANG BANGUN MODEL DAN PEMBAHASAN .....	23
4.1. Gambaran Umum .....	23

4.2. Perancangan <i>Model Platform Semi-Submersible</i> .....	23
4.3.1. Perancangan Sistem Mekanik Model.....	23
4.3.2. Perancangan Sistem Elektronik Model.....	25
4.3. Perancangan Sistem Dynamic Positioning .....	29
4.4. Diagram Blok Kontrol Sistem .....	31
4.5. Perancangan Sistem Kontrol Fuzzy .....	32
4.5.1. <i>Membership Function Input</i> .....	33
4.5.2. <i>Membership Function Output</i> .....	37
4.5.3. <i>Rule Based Fuzzy</i> .....	41
4.6. Respon Pergerakan .....	43
4.6.1. <i>Stand-by</i> .....	43
4.6.2. <i>Surge</i> .....	44
4.6.3. <i>Sway</i> .....	44
4.6.4. <i>Yaw</i> .....	45
4.6.5. Kombinasi <i>Surge</i> dan <i>Sway</i> .....	46
4.7. Kurva Hubungan Input dan Output Sistem Kontrol .....	47
4.7.1. Kurva Input Delta X dan Delta Y terhadap Output Servo .....	47
4.7.2. Kurva Input Delta X dan Delta Y terhadap Output Motor.....	50
4.7.3. Kurva Input Delta X dan Error Heading terhadap Output Motor dan Servo .	52
4.7.4. Kurva Input Delta Y dan Error Heading Terhadap motor dan Servo.....	54
4.8. Analisa Hasil Uji Coba .....	55
4.8.1. Uji Coba Manual Model .....	55
4.8.2. Uji Coba Otomatis Model.....	59
4.9. Validasi Alat .....	69
BAB V. KESIMPULAN DAN SARAN .....	72
5.1. Kesimpulan .....	73
5.2. Saran .....	73
DAFTAR PUSTAKA.....	75
LAMPIRAN A PROGRAM ARDUINO BASE REFERENCE.....	77
LAMPIRAN B PROGRAM STM KONTROL PLATFORM .....	85
LAMPIRAN C PROGRAM DESIGN FUZZY DI MATHLAB .....	123
LAMPIRAN D SPECIFIKASI.....	127
PROFIL PENULIS.....	129

## DAFTAR GAMBAR

Gambar 2.1 Dinamika Gerak Platform 3 DOF [4] .....	3
Gambar 2. 2 Position Reference System [3] .....	5
Gambar 2. 3 MODU <i>Semi Submersible Platform</i> [7].....	8
Gambar 2. 4 Modul <i>Microcontroler</i> STM 32 F407 [10].....	10
Gambar 2. 5 Modul <i>Microcontroler</i> Arduino Mega [12].....	10
Gambar 2. 6 Modul GPS [13] .....	10
Gambar 2. 7 Modul Kompas CMPS11 [14].....	11
Gambar 2. 8 Motor Servo [12] .....	12
Gambar 2. 9 Motor DC Brushless Watertight [15] .....	12
Gambar 2. 10 Electronic Speed Controller [11].....	13
Gambar 2. 11 Baterai Lithium Polymer [12].....	13
Gambar 2. 12 Modul HC-12 [16].....	14
Gambar 2. 13 <i>Diagram Block Fuzzy Logic Control</i> [17] .....	15
Gambar 2. 14 Fungsi Keanggotaan Segitiga Fuzzy [17].....	15
Gambar 3. 1 Alur Metodologi Penelitian .....	17
Gambar 3. 2 Skenario Uji Coba Pergerakan <i>Surge</i> .....	19
Gambar 3. 3 Skenario Uji Coba Pergerakan <i>Sway</i> .....	20
Gambar 3. 4 Skenario Uji Coba Pergerakan <i>Yaw</i> .....	20
Gambar 3. 5 Skenario Uji Coba Pergerakan Kombinasi <i>Surge</i> dan <i>Sway</i> .....	21
Gambar 4. 1 Desain Model <i>Platform Semi-submersible</i> . .....	23
Gambar 4. 2 Perhirungan <i>Hydrostatis</i> . .....	24
Gambar 4. 3 Model <i>Platform Semi-Submersible</i> .....	24
Gambar 4. 4 Implementasi Sistem Propulsi <i>Azimuth</i> .....	25
Gambar 4. 5 <i>Wiring Diagram</i> Rangkaian Elektronik pada Model.....	26
Gambar 4. 6 Skema Rangkaian Elektronik Pada <i>Model Semi-submersible</i> .....	27
Gambar 4. 7 Skema Rangkaian Elektronik Pada Based Refrence Point .....	28
Gambar 4. 8 Wiring Diagram Rangkaian Elektronik pada Based.....	28
Gambar 4. 9 Implementasi Based Reference Point .....	29
Gambar 4. 10 Sistem DPS dengan Metode DGPS .....	30
Gambar 4. 11 Rancangan Filter DGPS untuk mengurangi Error GPS .....	30
Gambar 4. 12 Diagram Block Dynamic Positioning System .....	32
Gambar 4. 13 Desain Kontrol Fuzzy DPS.....	33
Gambar 4. 14 Skema penentuan nilai error delta x dan delta y .....	34
Gambar 4. 15 Merbership Function Input Delta X.....	35
Gambar 4. 16 Membership Function Input Delta Y .....	36
Gambar 4. 17 Skema Penentuan Error Heading .....	36
Gambar 4. 18 Membership Function Error Heading .....	37
Gambar 4. 19 Data PWM Pergerakan Servo.....	38
Gambar 4. 20 Membership Fuction Servo 1 .....	39
Gambar 4. 21 Membership Function Servo 2 .....	39

Gambar 4. 22 Membership Function Servo 3.....	39
Gambar 4. 23 Membership Function Servo 4.....	39
Gambar 4. 24 <i>Membership Function</i> Motor 1 .....	40
Gambar 4. 25 <i>Membership Function</i> Motor 2 .....	40
Gambar 4. 26 <i>Membership Function</i> Motor 3 .....	41
Gambar 4. 27 <i>Membership Function</i> Motor 4 .....	41
Gambar 4. 28 Rule Based DPS Kontrol .....	41
Gambar 4. 29 Skema Pergerakan Model .....	43
Gambar 4. 30 Respon posisi arah hadap motor kondisi <i>Stand-by</i> .....	43
Gambar 4. 31 Respon posisi arah hadap motor kondisi <i>Surge</i> .....	44
Gambar 4. 32 Respon posisi arah hadap motor kondisi <i>Sway</i> .....	45
Gambar 4. 33 Respon posisi arah motor kondisi <i>Yaw</i> .....	45
Gambar 4. 34 Respon posisi arah motor kondisi quarter.....	46
Gambar 4. 35 Kurva Delta X dan Delta Y terhadap Servo 1 .....	47
Gambar 4. 36 Kurva Delta X dan Delta Y terhadap Servo 2 .....	48
Gambar 4. 37 Kurva Delta X dan Delta Y terhadap Servo 3 .....	48
Gambar 4. 38 Kurva Delta X dan Delta Y terhadap Servo 4 .....	49
Gambar 4. 39 Kurva Delta X delta Y terhadap Motor 1 .....	50
Gambar 4. 40 Kurva Delta X dan Delta Y terhadap Motor 2.....	50
Gambar 4. 41 Kurva Delta X dan Delta Y terhadap Motor 3.....	51
Gambar 4. 42 Kurva Delta X dan Delta Y terhadap Motor 4.....	52
Gambar 4. 43 Kurva Delta X dan Error Heading terhadap Motor.....	52
Gambar 4. 44 Kurva Delta X dan eror Heading terhadap Servo 4 .....	53
Gambar 4. 45 Kurva Delta Y dan Error Heading terhadap Motor 4.....	54
Gambar 4. 46 Kurva Delta Y dan Error Heading terhadap Servo 4. ....	54
Gambar 4. 47 Uji Coba sistem Propulsi dan Elektronika Model .....	55
Gambar 4. 48 Pengujian Pergerakan Model .....	55
Gambar 4. 49 Penentuan Set Point dan Heading Platform.....	60
Gambar 4. 50 Penampakan lintasan ujicoba.....	69
Gambar 4. 51 Penampakan sistem monitoring platform dari darat .....	69
Gambar 4. 52 Monitoring Pergerakan Platform .....	70
Gambar 4. 53 Kondisi Nyata Pergerakan Platform .....	70
Gambar 4. 54 Monitoring Error Jarak Platform Terhadap Titik Tujuan.....	71
Gambar 4. 55 Monitoring Error Sudut Platform Terhadap Heading .....	71

## DAFTAR TABEL

Tabel 2. 1 Perbandingan Notasi Class DPS di ABS.....	5
Tabel 4. 1 Dimensi Model <i>Platform Semi Submersible</i> .....	25
Tabel 4. 2 Data PWM Pergerakan Servo Terhadap Sudut .....	38
Tabel 4. 3 Data PWM <i>Thruster</i> terhadap Kecepatan dan <i>Thrust</i> .....	40
Tabel 4. 4 Hasil Uji Coba Pergerakan Manual.....	56
Tabel 4. 5 Data Titik <i>Set Point</i> Dan Titik Refrensi Model.....	60
Tabel 4. 6 Data ujicoba pergerakan Surgee Heading 0° .....	60
Tabel 4. 7 Data respon kecepatan motor pergeseran maju .....	61
Tabel 4. 8 Data servo gerak mundur .....	61
Tabel 4. 9 Data respon kecepatan motor pergeseran mundur.....	61
Tabel 4. 10 Data servo gerak maju.....	62
Tabel 4. 11 Data ujicoba pergerakan Sway Heading 0° .....	62
Tabel 4. 12 Data respon kecepatan motor pergeseran kanan .....	63
Tabel 4. 13 Data servo gerak kiri .....	63
Tabel 4. 14 Data respon kecepatan motor pergeseran maju .....	63
Tabel 4. 15 Data servo gerak kanan. ....	64
Tabel 4. 16 Data ujicoba pergerakan Yaw Heading 0° .....	64
Tabel 4. 17 Data respon kecepatan motor pergeseran Yaw .....	65
Tabel 4. 18 Data servo gerak CCW.....	65
Tabel 4. 19 Data servo gerak CW .....	65
Tabel 4. 20 Data ujicoba pergerakan kombinasi Surge dan Sway Heading 0° .....	65
Tabel 4. 21 Data respon kecepatan motor pergeseran maju kanan.....	66
Tabel 4. 22 Data servo gerak mundur kiri.....	66
Tabel 4. 23 Data respon kecepatan motor pergeseran maju kiri.....	67
Tabel 4. 24 Data servo gerak mundur kanan.....	67
Tabel 4. 25 Data respon kecepatan motor pergeseran mundur kanan .....	67
Tabel 4. 26 Data servo gerak maju kiri .....	68
Tabel 4. 27 Data respon kecepatan motor pergeseran mundur kiri .....	68
Tabel 4. 28 Data servo gerak maju kanan .....	68

*“Halaman Ini Sengaja Dikosongkan”*



# BAB I. PENDAHULUAN

## 1.1. Latar Belakang

Pada era Revolusi Industri 4.0 saat ini dimana sistem otomasi dalam berbagai bidang aspek kehidupan menjadi hal yang paling utama khususnya dalam bidang Industri. Hal ini bertujuan untuk bisa meningkatkan nilai efektifitas dan efisiensi dalam berbagai aspek bidang pekerjaan Industri untuk menunjang produktifitas. Tak luput dari Revolusi Industri 4.0, bidang maritim pun turut andil didalamnya untuk pengembangan teknologi maritim yang lebih efektif dan efisien.

Indonesia sebagai salah satu negara dengan kegiatan maritim yang sangat tinggi sudah seharusnya ikut mengembangkan teknologi Industri yang berbasis revolusi industri 4.0. Tidak dapat dipungkiri bahwa kekayaan maritim Indonesia yang sangat melimpah berupa lautan yang luas, garis pantai yang sangat panjang, kekayaan alam berupa ikan, minyak bumi, gas alam dan lain sebagainya. Hal inilah yang membuat harus adanya pengembangan teknologi otomasi di bidang maritim untuk bisa meningkatkan produktifitas agar lebih efisien dan efektif. Salah satu kegiatan maritim di Indonesia yang sangat tinggi aktifitasnya adalah kegiatan Eksplorasi dan Produksi minyak dan gas di area lepas pantai dengan menggunakan kapal FPSO ataupun dengan menggunakan bangunan lepas pantai.

Pada umumnya proses Eksplorasi dan Produksi minyak dan gas di area lepas pantai pada kedalaman 300 meter menggunakan *platform* terapung seperti *borship* dan *platform semi submersible*. Untuk menunjang proses Eksplorasi dan Produksi, *platform* tersebut harus tetap stasionary pada lokasi yang diinginkan. Umumnya *Platform* tersebut dipancang dengan menggunakan 8 - 12 anchor untuk bisa menahan berbagai hambatan beban baik yang diberikan oleh beban pada *platform* itu sendiri ataupun dari kondisi lingkungan laut seperti arus, ombak dan badai [1].

Permintaan akan sistem eksplorasi dan produksi terapung berupa Platform Semi-Submersible pada laut dalam saat ini pun semakin meningkat. Semi-Submersible adalah teknologi platform eksplorasi dan produksi yang dapat mudah beradaptasi di laut dalam dan merupakan salah satu sistem produksi terapung lepas pantai yang paling efektif dan efisien. Semi-Submersible dapat dirancang secara khusus untuk bisa melakukan kegiatan eksplorasi dan produksi hingga pada kedalaman laut 1000 meter. Hal ini membuat Platform Semi-Submersible sangat tidak memungkinkan menggunakan sistem mooring dan anchoring untuk bisa mempertahankan posisinya saat melakukan kegiatan eksplorasi pengeboran. Karena dalam proses kegiatan eksplorasi khususnya pengeboran, platform FPSO ini harus tetap bisa mempertahankan posisinya. [2]

Dengan kondisi laut yang sangat dalam tersebut membuat sistem tabat dengan anchor yang umumnya dipasang di area dengan kedalaman 100 m menjadi tidak praktis dan ekonomis lagi untuk digunakan. Dengan kondisi tersebut, penggunaan *Dynamic Positioning System* bisa menjadi salah satu solusi untuk bisa mengatasi masalah *platform* untuk tetap bisa stasionary pada lokasi yang diinginkan.

DPS ini merupakan salah satu inovasi teknologi sistem otomasi yang dapat menunjang kegiatan eksplorasi dan produksi minyak dan gas di area lepas pantai lebih efektif, ekonomis dan efisien. Hal itulah yang melatar belakangi untuk bisa mendesain rancang bangun sistem dynamic positioning sistem pada semi submersible platform untuk turut mengembangkan teknologi maritim di Indonesia dengan berbasis Otomasi.

### **1.2. Rumusan Masalah**

Adapun rumusan masalah dalam rancang bangun Dynamic Positioning System in antara lain :

- 1) Bagaimana cara mengontrol DPS pada *Model Semi-Submersible Platform* menggunakan *logika Fuzzy*?
- 2) Bagaimana pengaruh jarak dan arah perpindahan platform terhadap kecepatan motor dan respon pergerakan dari *Model Semi-Submersible Platform*?

### **1.3. Tujuan**

- 1) Merancang kontrol DPS secara otomatis pada Model Semi-Submersible Platform dengan menggunakan Logika Fuzzy.
- 2) Mengetahui pengaruh jarak dan arah perpindahan platform terhadap kecepatan motor dan respon pergerakan dari Model Semi-Submersible Platform.

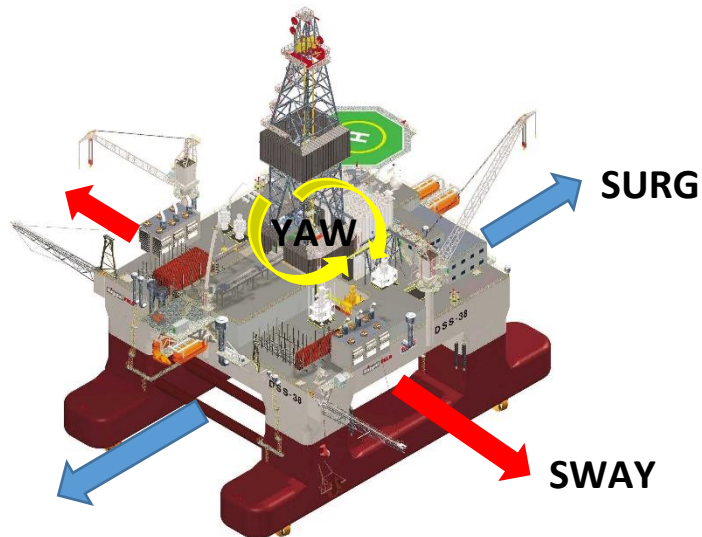
### **1.4. Batasan Masalah**

- 1) Dalam perancangan DPS pada *Platform Semi-Submersible* ini dilakukan untuk mengatasi masalah positioning pada 3 sumbu derajat kebebasan yaitu *Yaw, Surge, Sway*.
- 2) Pengujian diasumsikan tanpa adanya gangguan berupa arus dan gelombang.
- 3) Koefisien dan *hydrodynamic force* pada rancang bangun Model Platform Semi-submersible ini tidak dianalisa.

## BAB II. STUDY LITERATUR

### 2.1. *Dynamic Positioning System*

Dynamic Positioning System merupakan sebuah sistem yang di kontrol secara otomatis untuk mempertahankan posisi sebuah kapal atau platform dengan menggunakan sistem propulsi. Definisi ini mencakup tidak hanya untuk mempertahankan posisi saja, namun sistim *Dynamic Position* juga sangat mendukung agar sebuah kapal atau platform bisa melakukan manufer secara presisi, tracking dalam perpindahan posisi, dan kemampuan positioning yang lainnya [3].



Gambar 2.1 Dinamika Gerak Platform 3 DOF [4]

Setiap kapal atau semisubs platform dapat bergerak dalam 6 derajat kebebasan, meliputi 3 jenis pergerakan rotasi yaitu *Yaw*, *Pitch*, *Roll*. Dan 3 jenis pergerakan translasi yaitu, *Surgee*, *Sway*, dan *heave*. Sistem *Dynamic Positioning* secara konvensional bekerja untuk mengkontrol secara otomatis 3 jenis pergerakan yaitu *Surgee*, *Sway* dan *Yaw*. Pergerakan *Surgee* dan *Sway* berhubungan dengan positioning platform terhadap titik koordinat, sedangkan *Yaw* berhubungan dengan arah hadap platform tersebut.

DPS dapat dibagi menjadi 4 sub-sistem utama, yaitu : Sistem Sensor, Controller, Thruster System, serta Power System [5]. Sistem sensor berfungsi untuk mengukur nilai dari setiap pergeseran platform yang dipengaruhi oleh kondisi lingkungan. kontroller menjadi pusat utama kontrol otomatis ataupun manual untuk mengolah data yang didapatkan dari sensor untuk bisa kembali mempertahankan posisi platform. Thruster sistem berfungsi sebagai penggerak utama platform untuk memberikan gaya perlawanan terhadap pergeseran platform akibat gaya dari luar. Dan sistem power merupakan sistem energi utama pada platform untuk bisa beroperasi sebagaimana mestinya.

Setiap vessel di laut selalu dihadapkan dengan gaya-gaya yang ada di laut seperti angin, gelombang dan arus laut yang harus dapat diatasi oleh vessel tersebut melalui sistem penggeraknya. Hal itu dikarenakan gaya-gaya tersebutlah yang mempengaruhi posisi dan arah hadap dri vessel atau platform tersebut. Dengan adanya gaya-gaya

tersebut, platform harus dapat mengontrol posisi dan arah hadapnya sesuai dengan batas toleransi untuk tetap mempertahankan posisinya. Sistem dynamic position mempertahankan posisinya dan arah hadapnya dengan cara menghitung nilai offset perpindahan dari posisi awal yang disebabkan oleh gaya dari luar kemudian akan menghitung kekuatan yang harus dihasilkan oleh thruster untuk mengurangi kesalahan sampai mendekati nol.

Beberapa keuntungan dari sistem Dynamic Positioning adalah : Platform sepenuhnya self-propelled, pengaturan posisi platform lebih mudah dan cepat, kapal sangat mudah dalam bermanuver, respon terhadap perubahan cuaca cepat, fleksibilitas dalam sistem, mampu beroperasi dikondisi laut dalam, dapat menyelesaikan tugas-tugas lebih cepat, sehingga lebih ekonomis, menghindari risiko merusak perangkat keras dasar laut jika menggunakan tambat dan jangkar, dapat menghindari lokasi tambat kapal atau platform lainnya dan platform dapat berpindah ke lokasi baru dengan cepat [3].

### 2.1.1. Classification Notation

Berdasarkan klasifikasi ABS [6] tentang *Dynamic Positioning System* pada *vessels*, sistem *Dynamic Positioning* memiliki beberapa pengelompokan notasi sesuai dengan klasifikasi, antara lain :

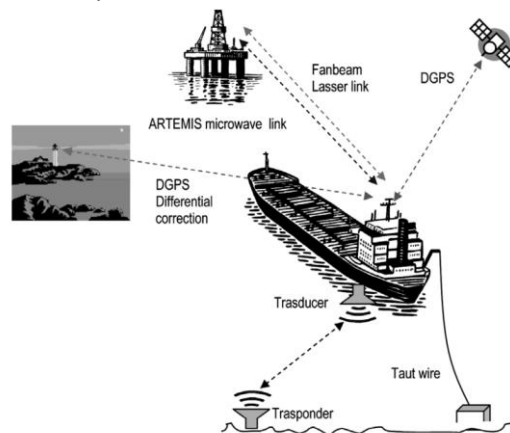
- a. DPS-0  
Untuk kapal, yang dilengkapi dengan kontrol posisi manual terpusat dan sistem kontrol hadap otomatis untuk mempertahankan posisi dan arah hadap di bawah kondisi lingkungan yang maksimum.
- b. DPS-1  
Untuk kapal, yang dilengkapi dengan sistem penentuan posisi dinamis yang mampu secara otomatis mempertahankan posisi dan arah kapal di bawah kondisi lingkungan maksimum yang ditentukan memiliki sistem kontrol posisi manual.
- c. DPS-2  
Untuk kapal, yang dilengkapi dengan sistem penentuan posisi dinamis yang mampu secara otomatis mempertahankan posisi dan arah kapal dalam lingkup operasi tertentu di bawah kondisi lingkungan maksimum yang ditentukan selama dan mengikuti setiap kesalahan tunggal, tidak termasuk hilangnya kompartemen atau kompartemen.
- d. DPS-3  
Untuk kapal, yang dilengkapi dengan sistem pemosisian dinamis yang mampu secara otomatis mempertahankan posisi dan taju kapal dalam lingkup operasi tertentu di bawah kondisi lingkungan maksimum yang ditentukan selama dan mengikuti setiap kesalahan tunggal, termasuk hilangnya kompartemen sepenuhnya akibat kebakaran. atau banjir.

Berikut merupakan tabel perbandingan dari sistem *Dinamic Positoning System* pada masing-masing notasi :

Tabel 2. 1 Perbandingan Notasi Class DPS di ABS

	DPS-0 (Class 0)	DPS-1 (Class 1)	DPS-2 (Class 2)	DPS-3 (Class 3)
Independent, Manual Thruster Controls <sup>1</sup>	1	1	1	1
Joystick System w/ AutoHeading <sup>4</sup>	0	1	1	1
Automatic System (DP)	0	1	2	2+1
Gyrocompasses	1	2	2	3 <sup>2</sup>
Wind Sensors	1	2	2	2
Motion Reference Sensors	1	1	2	3
Position Reference Systems	1	2	3 <sup>5</sup>	3 <sup>2,5</sup>
UPS w/ 30 min. battery	0	1	2	3
Consequence Analyzer	0	0	1	1
Power Management System <sup>3</sup>	0	0	1	1

### 2.1.2. Position Reference System



Gambar 2. 2 Position Reference System [3]

Semua kapal atau platform yang menganut system dynamic position system pasti memiliki PRS (*Position Reference System*). Ada lima jenis PRS yang biasa digunakan pada platform dengan sistem dynamic position :

#### a. *Hydroacoustic Position Reference (HPR)*

Hydroacoustic Position Reference terdiri dari satu atau lebih transponder yang berada di dasar laut dan transducer yang ditempatkan di lambung platform. Prinsip kerja dari sistem ini adalah dengan cara transducer mengirim sinyal akustik ke transponder. Jarak antara transducer dan transponder dapat diketahui dengan membaca kecepatan suara akustik yang dipancarkan transducer dalam air. Arah sinyal akustik dari transponder akan dapat diketahui oleh transducer karena terdapat banyak element pengukuran di dalam transducer tersebut. Dengan demikian posisi platform dapat diketahui dan dihitung untuk melakukan respon [3].

Kerugian dari sistem HPR ini adalah kerentanan terhadap gangguan suara lainnya yang dapat dibaca oleh transducer, seperti suara bising thruster, gelombang, dan suara akustik lainnya.

b. *Taut Wire*

Sistem taut wire ini sangat berguna untuk referensi posisi jarak pendek, terutama dimana kapal atau platform berada di suatu posisi tersebut cukup mala dan dalam kondisi laut yang tidak terlalu dalam. Dengan taut wire ini platform atau kapal tidak hanya bisa mempertahankan posisi statisnya tetapi juga dapat mempertahankan posisi relatif terhadap benda lain yang bergerak.

Prinsip kerja dri taut wire ini adalah dengan menggunakan crane yang terpasang di deck platform. Untuk sistem tarik ulur kawat tersebut ditangani oleh winch dengan tegangan konstan. Di ujung boom crane terdapat sensor sudut yang akan mendeteksi sudut kawat. Ketika depressor diturunkan ke dasar laut, sistem akan beralih ke mode tambat dan winch akan bekerja mempertahankan tegangan kawat agar tetap konstan sehingga dengan demikian dapat mendeteksi pergerakan platform jika terdapat perubahan tegangan. Keakuratan sistem DP pada Taut Wire tergantung pada kedalaman air, ketegangan tambat, sudut kawat ke vertikal dan kekuatan pemasangannya.

Sistem taut wire memiliki keterbatasan terhadap sudut kawat, diaman semakin meningkatnya sudut kawat maka akan semakin meningkat juga resiko terseetnya beban yang disangkutkan. Dengan demikian sudut yang diijinkan maksimal adalah 30 derajat. Pada sudut 35 derajat, alarm akan berbunyi dan sistem DP akan menolak referensi posisi tersebut [3].

c. *Differential Global Positioning (DGPS)*

*Global Positioning System* (GPS) memiliki keakuratan sesuai standard layanan positioning GPS sebesar 20m, tetapi akurasi GPS tidak memadai untuk tujuan DP. Untuk meningkatkan akurasi GPS, koreksi diferensial diterapkan pada data GPS. Hal ini dilakukan dengan membangun stasiun referensi pada titik yang diketahui pada spheroid WGS 84. Rentang semu yang diperoleh oleh receiver dibandingkan dengan yang dihitung dari lokasi satelit yang diketahui, dan stasiun referensi, mendapatkan koreksi [3].

Sebagian besar layanan DGPS menerima berbagai masukan diferensial yang diperoleh dari berbagai stasiun referensi yang terpisah secara luas. Sistem DGPS memberikan stabilitas dan akurasi yang lebih besar, dan menghilangkan lebih banyak kesalahan ionosfer daripada yang dapat diperoleh dari satu stasiun referensi. Akurasi yang diperoleh dari sistem DGPS adalah sekitar 1-3m tergantung pada jarak ke stasiun referensi, kondisi ionosfir, dan konstelasi satelit yang tersedia.

d. *Laser Based System (Fanbeam and CyScan)*

Ada dua referensi posisi DP laser yang digunakan: *Fanbeam* dan *CyScan*. Kedua sistem mengunci ke satu target dan / atau sejumlah target pada struktur, dari posisi mana harus dipertahankan. Pulsa dikirim dan diterima sehingga rentang dan bantalan dapat diukur.

Sistem *Fanbeam* diproduksi oleh MDL dari Aberdeen dan terdiri dari dua unit, *laser scanner* dan *Universal Display Unit* (UDU). Laser ini terdiri dari array gallium arsenide semikonduktor laser dioda. Hal ini menghasilkan laser dengan kipas vertikal 20° dengan perbedaan horizontal kurang dari 4 miliradians, berdenyut pada 5.000 Hz. Pulsa yang dipancarkan melalui lensa pemancar. Lampu diterangi diterima diarahkan ke array dioda sensitif foto untuk

menghasilkan sinyal listrik. Jarak ditentukan dari gema mulai; akurasi ditingkatkan dengan rata-rata sejumlah pengembalian dari target. Ini adalah sistem yang sangat sederhana, karena hanya prisma kecil yang perlu dipasang pada struktur di dekatnya. Beberapa risiko adalah bahwa Fanbeam dapat dikunci pada objek lain yang mencerminkan dan memblokir sinyal. Unit pemindai dalam bejana DP harus ditempatkan di lokasi yang memiliki pandangan tak terhalang dari cakrawala. Rentang bervariasi sesuai dengan kondisi cuaca, ketika sistem akan dipengaruhi oleh visibilitas optik berkurang. Itu biasanya lebih dari 500 meter [3].

e. *Artemis*

*Artemis* adalah nama dagang untuk sistem yang diproduksi oleh *Christian Huygenslaboratorium BV* di Belanda. Landasan posisi oleh *Artemis* adalah mendapatkan kisaran dan bantalan stasiun bergerak dari beberapa lokasi tetap yang diketahui. Tautan microwave daya rendah didirikan antara dua stasiun, menggunakan antena pelacakan terarah.

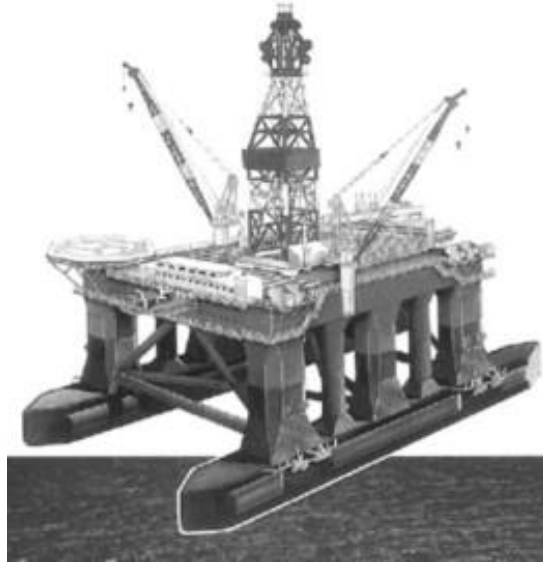
Penundaan sinyal berbanding lurus dengan jarak antara stasiun tetap dan stasiun bergerak. Biasanya, stasiun tetap terletak di atas landasan platform atau lokasi tetap lainnya, sedangkan ponsel terletak di kapal. Stasiun bergerak bertindak sebagai Master; kontrol dan masukan operator berasal dari ponsel. Unit seluler dihubungkan ke sistem DP.

Koreksi harus diterapkan dalam algoritma DP untuk memungkinkan perbedaan lokasi antara antena dan pusat putaran kapal.

Sistem ini terdiri dari dua Unit Dasar *Artemis* yang identik, yang dikonfigurasi sebagai unit tetap dan sebagai unit bergerak. Antarmuka operator disediakan dengan menggunakan Panel Operator Diperpanjang (EOP) di stasiun bergerak, dan oleh Panel Operator Dasar (BOP) di Stasiun Tetap. EOP adalah antarmuka pengguna di atas kapal dan dengan nyaman dipasang di dekat konsol operator DP, atau terintegrasi di dalamnya.

Antena terdiri dari pelacakan gelombang udara, dipasang dalam dua bagian. Antena pelacak memiliki lebar balok horizontal  $2^\circ$  dan lebar balok vertikal sekitar  $22^\circ$ . Energi terpancar adalah sekitar 100 mW, terpolarisasi vertikal. Ini diatur bahwa antena di stasiun tetap dan bergerak menjaga sambungan radio dengan melacak antena di azimuth. Tanpa mempertimbangkan pergerakan bejana, kedua antena tersebut saling berhadapan, dengan antena normal ke arah sinyal. Sisa dan paruh kanan antena menerapkan sinyal secara terpisah ke empat port. Sinyal diterapkan sama tetapi berlawanan dalam fase. Ini berarti bahwa output akan menjadi nol hanya jika antena tegak lurus terhadap arah sinyal yang masuk. Jika ini tidak terjadi, motor penggerak akan bertindak untuk memutar antena untuk mengurangi kesalahan tegak lurus ke nol. Dengan ini berarti kedua antena terus saling melacak satu sama lain [3].

## 2.2. *Semi-Submersible Platform*



Gambar 2. 3 MODU *Semi Submersible Platform* [7]

*Semi submersible platform* adalah platform khusus yang digunakan dalam sejumlah peran lepas pantai tertentu seperti sebagai rig pengeboran, kapal penyelamat, platform produksi minyak serta sebagai derek untuk mengangkat muatan berat atau platform lainnya [8].

Platform semi-submersible memperoleh sebagian besar daya apungnya dari sistem ballast yang terdapat dalam ponton. Ponton kedap air tersebut terletak dibagian bawah platform yang bisa tenggelam dan juga bisa terapung. Itulah mengapa platform ini disebut semi submersible. Dengan sistem ponton yang dapat tenggelam ini membuat platform memiliki stabilitas yang tinggi sehingga deck operasi pada platform bisa diletakkan lebih tinggi [4].

Dengan kondisi rancangan lambung semi submersible yang terendam, membuat jenis platform ini lebih sedikit menerima dampak beban gelombang dibandingkan dengan jenis kapal lain. Namun dengan luasan permukaan basah yang cukup kecil membuat platform semisubmersible sensitive terhadap perubahanbeban yang terdapat di platform. Sehingga harus lebih berhati-hati dalam pemindahan beban yang terdapat di platform untuk tetap menjaga stabilitasnya.

Platform semisubmersible dapat beroperasi di laut dalam ataupun dangkal dengan memanfaatkan sistem ballastingnya yang dapat merubah sarat platform menjadi lebih dalam atau lebih rendah. Adapun beberapa jenis platform semi-submersible sesuai fungsinya adalah sebagai berikut [8] :



a. *Mobile Offshore Drilling Units (MODU)*

MODU merupakan platform yang dibuat dengan stabilitas tinggi untuk kepentingan pengeboran minyak dan gas di area lepas pantai. Platform ini dapat ditarik ke posisi pengeboran dengan menggunakan tugboat yang menariknya ataupun bisa juga dengan menggunakan azimuth thruster yang terpasang di platform itu sendiri.

b. *Semi submersible Crane Vessel (SSCV)*

SSCV merupakan semisubmersible platform yang dilengkapi dengan dua crane besar yang digunakan sebagai vessel untuk pembangunan bangunan lepas pantai. Dengan sistem submersible dari platform inilah yang dimanfaatkan untuk tetap bisa menjaga stabilitas platform meskipun sedang mengangkat beban yang cukup berat.

c. *Offshore Support Vessel*

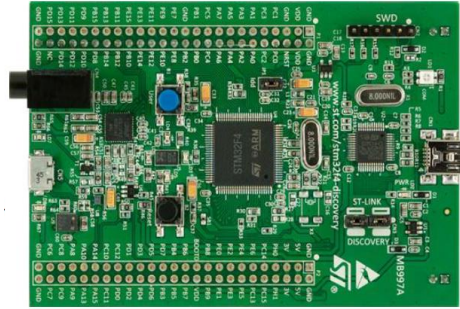
Merupakan semisubmersible yang digunakan sebagai support vessel dari bangunan lepas pantai lainnya karena memiliki stabilitas yang baik serta deck yang luas.

### **2.3. Azimuth Propulsion System**

Meskipun sistem pendorong azimuth merupakan inovasi yang sudah ditemukan lebih dari 50 tahun yang lalu, sistem ini masih tetap digunakan dengan berbagai macam keunggulannya. Azimuth propulsi sistem merupakan sistem propulsi dimana propeller dapat bergerak berputar 360 derajat terhadap sumbu vertikal. Dengan sistem ini secara otomatis akan meningkatkan kemampuan maneuver dari kapal atau platform yang menerapkannya. Dan juga tidak memerlukan lagi rudder serta reverse gear karena sistem propulsi azimuth sudah dilengkapi dengan reverse gear sendiri. Sistem ini terdiri dari beberapa komponen yang berbeda meliputi : azimuth thruster, sistem kontrol unit, dan shaft line [9].

### **2.4. Microcontroller STM32 F407**

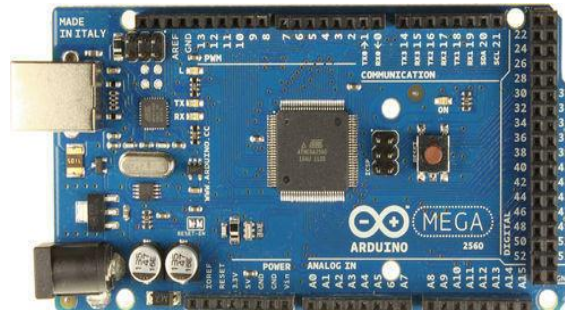
STM32 F407 merupakan salah satu mikrokontroler dengan jenis prosesor ARM 32 bit dengan tipe RISC yang sudah dilengkapi ST-LINK/V2 melalui koneksi USB. Mikrokontroler ini didesain dan diproduksi oleh STMicroelectronics sejak tahun 2007. Setiap mikrokontroler STM32 F407 terdiri dari inti prosesor, memori RAM statis, memori *flash*, *debugging interface*, dan berbagai periferal. Bahasa yang digunakan untuk memprogram mikrokontroler ini adalah bahasa pemrograman C yang umumnya banyak digunakan *engineer*. Di samping itu, dengan kebutuhan voltase 1.8-3.3 volt, mikrokontroler ini tergolong ke *low energy microcontroller* (www.st.com, 2011). Meskipun hanya membutuhkan energi yang kecil, *response time* dari mikrokontroler ini sangatlah cepat jika dibandingkan dengan mikrokontroler berprosesor Atmel AVR. Hal ini dikarenakan, prosesor ARM 32 Bit yang unggul di mayoritas lini prosesor mikrokontroler [10]. Bentuk board STM32F432 seperti Gambar di bawah ini.



Gambar 2. 4 Modul *Microcontroler* STM 32 F407 [10]

## 2.5. *Microcontroler* Arduino

Arduino merupakan sebuah mikrokontroler *single-board* yang bersifat *open-source*, diturunkan dari *wiring platform*, dirancang sedemikian hingga agar mudah digunakan dalam berbagai bidang. Arduino memiliki prosesor Atmel AVR dan memiliki bahasa pemrograman yang mudah diterjemahkan, bahasa pemrograman Arduino yang memiliki kemiripan *syntax* dengan bahasa pemrograman C. Arduino menggunakan keluarga mikrokontroler ATmega yang dirilis oleh Atmel sebagai basis. Pada kondisi lain terdapat *clone* arduino dengan menggunakan mikrokontroler berbeda dan tetap kompatibel dengan arduino pada level *hardware* [11]. Contoh gambar arduino ditunjukkan pada gambar berikut.



Gambar 2. 5 Modul *Microcontroler* Arduino Mega [12]

## 2.6. Sensor GPS



Gambar 2. 6 Modul GPS [13]

GPS merupakan navigasi yang memberikan informasi berupa posisi suatu objek di bumi, berbasis sistem satelit. Untuk dapat menentukan suatu posisi objek di bumi, sensor GPS membutuhkan 3 – 4 sinyal satelit yang ditangkap agar dapat membaca posisi dengan baik. Data yang didapat oleh sensor GPS berupa *longitude* dan *latitude*, selanjutnya dapat diketahui letak posisi, jarak, arah, dan kecepatan suatu objek pada posisi asal ke posisi tujuan dengan menggunakan perhitungan, dan bantuan berupa sensor kompas [13].

## 2.7. Sensor Kompas CMPS 11



Gambar 2. 7 Modul Kompas CMPS11 [14]

CMPS11 adalah kompas generasi ke 3 dengan sistem kompensasi kemiringan. Sensor kompas CMPS11 ini dilengkapi dengan magnetometer 3-axis, gyro 3-axis, dan akselerometer 3-axis. Filter Kalman menggabungkan gyro dan accelerometer untuk menghilangkan kesalahan yang disebabkan oleh memiringkan PCB. CMPS11 menghasilkan data sudut 0-3599 yang mewakili 0-359.9 atau 0 hingga 255 [14]. Output dari ketiga sensor adalah pengukuran terhadap nilai x, y, dan z terhadap medan magnet yang sekaligus untuk bisa mendapatkan nilai pitching dan rolling. Modul CMPS11 membutuhkan catu daya pada 3,6 - 5v dan menarik arus nominal 25mA. Pilihan antarmuka serial atau I2C disediakan. Sensor kompas ini digunakan sebagai inputan sudut arah hadap dari sebuah platform yang nantinya akan tetap dipertahankan pada kondisi dynamic positioning system [12].

## 2.8. Motor Servo

Motor servo adalah sebuah motor DC dengan sistem umpan balik tertutup di mana posisi rotor-nya akan diinformasikan kembali ke rangkaian kontrol yang ada di dalam motor servo. Motor ini terdiri dari sebuah motor DC, serangkaian gear, potensiometer, dan rangkaian kontrol. Potensiometer berfungsi untuk menentukan batas sudut dari putaran servo. Sedangkan sudut dari sumbu motor servo diatur berdasarkan lebar pulsa yang dikirim melalui kaki sinyal dari kabel motor servo.

Secara putaran terdapat dua jenis motor servo, yaitu motor servo standar dan motor servo continuous. Motor servo standar sering dipakai pada sistem robotika, misalnya untuk membuat Robot Arm (Robot Lengan) sedangkan motor servo continuous

sering dipakai untuk Mobile Robot. Pada bagian servo tertulis tipe servo yang bersangkutan. Contoh motor servo bisa dilihat pada Gambar 2.5.

Penggunaan motor servo di bidang robotika tentu ada alasannya. Pertama adalah motor servo memiliki putaran yang lambat dengan torsi yang kuat (berkat adanya sistem roda gigi). Hal ini cocok dengan bidang robotika.



Gambar 2. 8 Motor Servo [12]

## 2.9. Motor DC Brushless Watertight



Gambar 2. 9 Motor DC Brushless Watertight [15]

Motor Brushless adalah jenis motor yang memiliki konstruksi magnet permanen dan sebuah stator berkutub yang dililit kawat. Energi listrik diubah menjadi energi mekanik oleh pengaruh daya tarik menarik antara kekuatan magnet permanen dengan stator berkutub yang dililit kawat tembaga. Contoh motor brushless dapat dilihat pada Gambar 2.7. Bagian-bagian brushless motor adalah sebagai berikut:

- a. Stator  
Dasar dari stator brushless motor adalah sebuah stator dengan memiliki tiga buah gulungan.
- b. Rotor  
Rotor pada brushless motor terdiri dari beberapa magnet permanen. Jumlah kutub magnet di rotor juga mempengaruhi ukuran langkah dan riak torsi dari motor. Jumlah kutub yang banyak akan memberikan gerakan presisi dan riak

torsi yang kecil. Magnet permanen terdiri dari 1 sampai 5 pasang kutub. Dalam kasus tertentu bisa 8 pasang kutub.

### 2.10. ESC (*Electronic Speed Controller*)



Gambar 2. 10 Electronic Speed Controller [11]

Motor brushless memiliki sebuah ESC (*Elektronik Speed Control*) yang berfungsi sebagai pengatur kecepatan motor, selain itu juga berfungsi untuk menaikkan jumlah arus yang diperlukan oleh motor. Kecepatan untuk motor yang keluar dari ESC diatur melalui pulsa dari mikrokontroler [11].

### 2.11. Baterai *Lithium Polymer*



Gambar 2. 11 Baterai Lithium Polymer [12]

Baterai *lithium polymer* yaitu jenis baterai yang tidak menggunakan cairan sebagai elektrolit melainkan menggunakan elektrolit polimer kering yang berbentuk seperti lapisan plastik film tipis. Lapisan film ini disusun berlapis-lapis diantara anoda dan katoda yang mengakibatkan pertukaran ion. Dengan metode ini baterai *lithium polymer* dapat dibuat dalam berbagai bentuk dan ukuran. Diluar dari kelebihan arsitektur baterai *lithium polymer*, terdapat juga kekurangan yaitu lemahnya aliran pertukaran ion yang terjadi melalui elektrolit polimer kering. Hal ini menyebabkan penurunan pada *charging* dan *discharging rate*. Masalah ini sebenarnya bisa diatasi dengan memanaskan baterai sehingga menyebabkan pertukaran ion menjadi lebih cepat, namun metode ini dianggap tidak dapat untuk diaplikasikan pada keadaan sehari-hari [11].

### 2.12. Modul *Nirkabel HC-12*

Modul HC-12 adalah modul generasi terbaru yang memiliki multichannel jalur transmisi data. HC-12 bekerja pada pita frekuensi 433,4 sampai 473,0 MHz, dan beberapa

*channel* dapat diatur dengan loncatan 400kHz dari total 100 channel. Daya transmisi maksimum untuk modul ini adalah 100 mW (20 dBm), sensitivitas penerimaan adalah -117 dBm pada tingkat 5000 bps di udara. Modul ini mampu menempuh jarak 1000 meter (FU3 mode dengan kecepatan serial 4800 bps) di ruang terbuka, 1800 meter dengan FU4 mode.



Gambar 2. 12 Modul HC-12 [16]

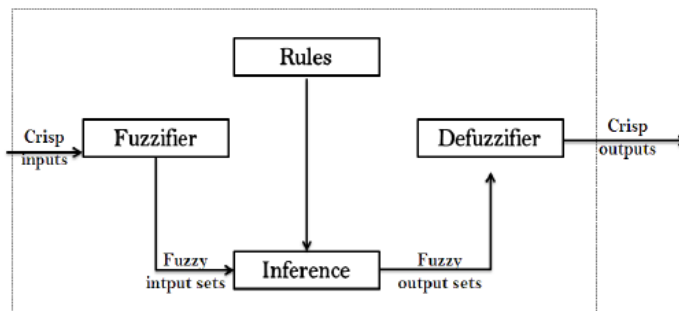
### 2.13. Coocox IDE (CoIDE)

Coocox adalah bahasa pemrograman yang tidak berbayar (freeware), digunakan untuk melakukan pemrograman untuk mikrokontroler ARM cortex Cortex M0, M3 dan M4. CoIDE ini menggunakan bahasa C yang merupakan bahasa yang umum digunakan dalam bahasa pemrograman. Selain itu CoIDE menawarkan fitur-fitur menarik sebagai sebuah IDE, seperti adanya komponen (atau library), code completion dan lain-lain.

### 2.14. Fuzzy Logic Control

Logika Fuzzy pertama kali dikembangkan oleh Lotfi A. Zadeh pada tahun 1965. Teori ini banyak diterapkan di berbagai bidang, antara lain representasi pikiran manusia kedalam suatu sistem. Banyak alasan mengapa penggunaan logika fuzzy ini sering dipergunakan antara lain, konsep logika fuzzy yang mirip dengan konsep berpikir manusia. Sistem fuzzy dapat merepresentasikan pengetahuan manusia ke dalam bentuk matematis dengan lebih menyerupai cara berpikir manusia. Pengontrol dengan logika fuzzy mempunyai kelebihan yaitu dapat mengontrol sistem yang kompleks, non-linier, atau sistem yang sulit di representasikan kedalam bentuk matematis. Selain itu, informasi berupa pengetahuan dan pengalaman mempunyai peranan penting dalam mengenali perilaku sistem di dunia nyata. [17]

Sistem kontrol fuzzy terdiri atas fuzzifier interface, interference of rules, rule base, dan defuzzifier Interface. Fuzzifier Interface menerjemahkan input sistem dari domain fuzzy dalam bentuk variable linguistik. Kemudian variable linguistik tersebut digunakan oleh interference of rules untuk menyimpulkan aturan IF-THEN yang disimpan dalam memori rule base. Rule Base sendiri adalah repositori dari semua pengetahuan yang digunakan untuk mengontrol sistem. Kemudian output dari interference of rules di bawa kembali ke defuzzifier interface seperti gambar berikut. [1]

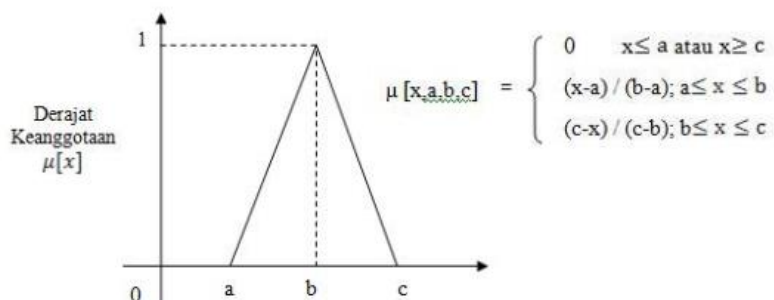


Gambar 2. 13 Diagram Block Fuzzy Logic Control [17]

Beberapa kelebihan yang dimiliki oleh logika fuzzy yaitu:

- Konsep logika fuzzy mudah dimengerti, dimana konsep matematis yang mendasari fuzzy sangat sederhana dan mudah dimengerti.
- Logika fuzzy bekerja didasarkan pada bahasa alami.
- Logika fuzzy memiliki toleransi terhadap data-data yang tidak tepat.
- Logika fuzzy mampu memodelkan fungsi-fungsi nonlinier yang sangat kompleks.
- Logika fuzzy dapat bekerja sama dengan teknik-teknik kendali secara konvensional

Berikut merupakan gambaran fungsi keanggotaan fuzzy dengan jenis keanggotaan segitiga seperti yang akan digunakan.



Gambar 2. 14 Fungsi Keanggotaan Segitiga Fuzzy [17]

*“Halaman Ini Sengaja Dikosongkan”*



# BAB III. METODOLOGI PENELITIAN

## 3.1. Metode Penelitian



Gambar 3. 1 Alur Metodologi Penelitian

Metodologi penelitian dalam perancangan sistem *Dynamic Position* pada Model platform semisubmersible adalah dengan cara melakukan perancangan sistem mekanisme dan algoritma penerapan *Dynamic Positioning* pada platform *Semi-Submersible*. Dalam desain perancangan ini akan menggunakan dua jenis type sistem propulsi yaitu sistem propulsi fiks dan sistem propulsi *azimuth*. Setelah dilakukan perancangan dengan melakukan simulasi, metode selanjutnya yang akan diterapkan adalah pembuatan Model yang kemudian akan dilakukan ujicoba secara berkala pada berbagai kondisi yang diharapkan sampai bisa mendapatkan sesuai dengan hasil yang dikehendaki.

### 3.2 Study Literatur

Tahapan awal dalam proses rancang bangun sistem *Dynamic Positioning System* adalah Studi Literatur. Studi literatur terdiri atas proses penelitian yang telah dilakukan sebelumnya mulai dari pengumpulan data hingga proses pengaplikasian. Selain proses penelitian, studi literatur juga mencakup analisa dari hasil pengaplikasian yang sudah pernah dilaksanakan. Output dari studi literatur mengenai rancang bangun Sistem Dinamic Position adalah sebuah prosedur kerja penelelitian dan daftar data yang dibutuhkan pada saat menemukan studi kasus.

### 3.3 Penentuan Studi Kasus dan Pengumpulan Data

Penentuan studi kasus dan pengumpulan data merupakan tahapan selanjutnya dalam proses rancang bangun sistem dynamic positioning system. Pada tahap ini bertujuan untuk menentukan arah fokus dalam penelitian serta untuk mengumpulkan semua data yang dibutuhkan dalam studi kasus tersebut untuk kemudian dijadikan bahan dalam proses perancangan dan pendesainan sistem DPS pada *Model Platform Semi-Submersible*.

### 3.4 Desain System

Desain sistem merupakan proses pembuatan rencana serta penggambaran rancangan sistem baik dalam bentuk gambar dan program. Proses ini terbagi menjadi dua yaitu desain sistem mekanisme DPS yang meliputi desain Model platform beserta sistem mekanisme propulsi yang ada di dalamnya yaitu sistem propulsi azimuth. Dan yang kedua adalah Desain sistem elektronik serta algoritma pemrograman DPS yang akan diterapkan pada *Model platform Semi-Submersible*.

#### a. Desain Mekanik

Adapun dalam desain mekanik ini yang akan dedesain untuk pembuatan Model Semi-submersible Platform terdiri dari :

##### 1. Desain Model

Desain Model akan disesuaikan dengan data contoh platform pembanding yang diskalakan menyesuaikan dengan tempat pengujian yang akan dilakukan. Rancangan desain dari *Model Semi-submersible Platform* akan di disain dengan menggunakan aplikasi *Maxsurf*.

##### 2. Desain Mekanisme *Azhimut Propulsion*

Merupakan proses mendesain sistem mekanisme azimuth propulsion yang akan di terapkan di Model. Dimana dalam Model akan menggunakan 4 motor azimuth thruster dan 4 motor servo sebagai rotator dari masing2 motor thruster.

b. Desain Elektronik

Dalam desain elektronik ini akan terfokus pada desain rangkaian sistem kontrol, online diagram sistem kontrol, serta desain diagram blok dari sistem kontrol yang akan diterapkan pada Model.

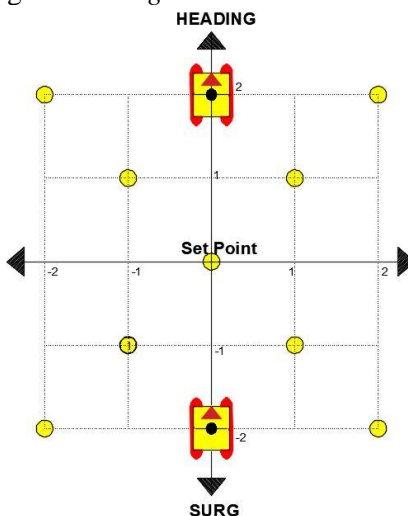
### 3.5. Pembuatan Model dan Program Sistem Kontrol

Setelah dilakukannya proses desain tahapan selanjutnya adalah proses perancangan baik sistem mekanismenya dan sistem kontrol yang kemudian akan dikombinasikan dan diterapkan dalam pembuatan Model platform semisubmersible sesuai dengan desain yang telah dibuat. Dalam perancangan program sistem kontrol DPS pada Model akan menggunakan sistem kontrol Fuzzy yang dimana data yang akan diterapkan pada Model.

### 3.6. Skenario Uji Coba Model

Pengujian Model platform *Semi-submersible* dengan *Dynamic Positioning System* dilakukan sesuai dengan cara melakukan pergeseran pada prototipr terhadap set point yang sudah ditentukan dengan berbagai macam arah sehingga dapat mengetahui respon pergerakan platform apakah sudah sesuai dengan algoritma yang dirancang. Terdapat empat skenario utama pada pengujian Model untuk mengetahui respon pergerakan serta pengaruh jarak perpindahan terhadap kecepatan motor. Skenario tersebut meliputi dari jenis pergeseran yang dilakukan yaitu pergerakan *Surge*, *Sway*, *Yaw*, dan kombinasi *Surge* dengan *Sway*.

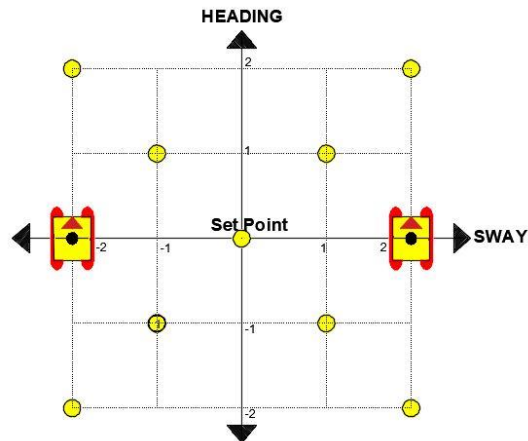
a. Skenario Uji Coba Pergerakan *Surge*



Gambar 3. 2 Skenario Uji Coba Pergerakan *Surge*

Pada skenario uji coba ini Model platform *Semi-submersible* akan digeser sejauh 2 m pada arah pergerakan *Surge* dari Set Point atau titik tujuan platform. Skenario pergerakan *Surge* terdiri dai 2 pergeseran yaitu pergeseran Maju terhadap set point dan pergeseran mundur terhadap set point. Dari pergeseran tersebut kemudian platform akan dicoba secara otomatis untuk mengetahui respon dari pergeseran Model sehingga bisa kembali menuju ke set point.

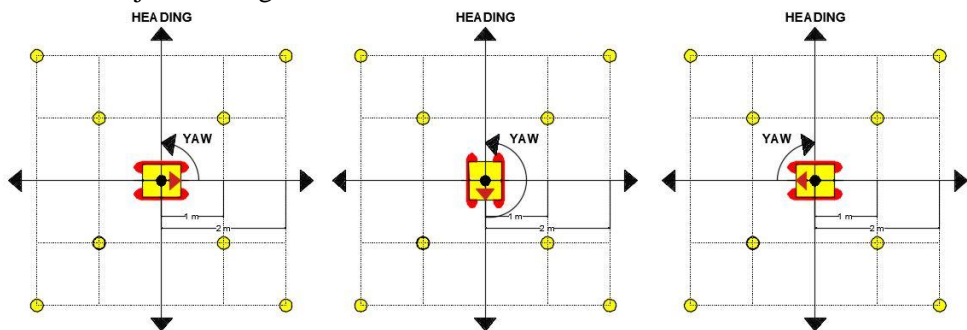
b. Skenario Uji Coba Pergerakan *Sway*



Gambar 3. 3 Skenario Uji Coba Pergerakan *Sway*

Pada skenario uji coba ini Model platform *semi-submersible* akan digeser sejauh 2 m pada arah pergerakan *Sway* dari *Set Point* atau titik tujuan platform. Skenario pergerakan sway terdiri dari 2 macam pergeseran yaitu pergeseran ke kanan set point dan pergeseran ke kiri set point. Dari pergeseran tersebut platform kemudian akan diuji secara otomatis untuk mengetahui respon pergerakan platform agar dapat kembali ke set point.

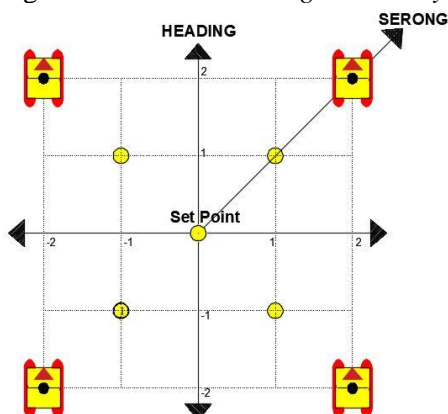
c. Skenario Uji Coba Pergerakan *Yaw*



Gambar 3. 4 Skenario Uji Coba Pergerakan *Yaw*

Pada Skenario ujicoba pergerakan *Yaw* platform akan tetap berada diposisi set point namun arah hadap platform akan digeser ke 3 sudut yang berbeda dari sudut heading platform. Terdapat 3 perputaran sudut yang akan dicoba pada pergerakan ini yaitu 90 derajat, 180 derajat dan 270 derajat terhadap heading platform. Setelah diputar ke arah sudut yang dituju platform akan dijalankan secara otomatis untuk mengetahui respon platform agar bisa kembali menghadap ke sudut heading awal platform.

d. Skenario Uji Coba Pergerakan Kombinasi *Surge* dan *Sway*



Gambar 3. 5 Skenario Uji Coba Pergerakan Kombinasi *Surge* dan *Sway*

Pada skenario ujicoba ini platform akan digeser dengan arah serong sejauh 2 m terhadap set point atau titik tujuan platform. pada ujicoba ini terdapat 4 macam pergeseran platform secara serong yaitu pergeseran maju kanan, maju kiri, mundur kanan, dan mundur kiri. Setelah dilakukan penggeseran platform akan dijlankan secara otomatis untuk mengetahui respon agar platform bisa kembali menuju ke *set point* yang sudah ditentukan.

### 3.7. Analisa Hasil Uji Coba dan Perhitungan Data

Dari hasil pengujian pada tahap sebelumnya akan didapatkan beberapa data uji coba yang kemudian akan dianalisa untuk mengetahui apakah hasil pengujian dari Model pada kolam arus tersebut sudah sesuai dengan hasil yang diharapkan atau belum. Jika nilai eror yang dihasilkan dari proses ujicoba masih cukup besar maka akan dilakukan proses pengkajian ulang dalam proses perancangan sistem mekanisme dan algoritma pemrograman hingga nilai eror yang dihasilkan mendekati nol.

Adapun variable data yang akan dikelola adalah mengenai respon pergerakan platform terhadap perpindahan platform sesuai dengan 4 macam skenario pergerakan yang sudah ditentukan. Sehingga data yang akan diambil meliputi data bacaan sensor GPS dan kompas, RPM motor, jarak pergeseran serta respon waktu terhadap setiap kondisi skenario pergerakan.

### 3.8. Validasi

Validasi merupakan tahapan dimana hasil ujicoba *Dynamic Positioning System* dari Model platform *Semi-submersible* pada lokasi pengujian dibandingkan dengan hasil monitoring pergerakan platform dari darat dengan menggunakan sistem *remote desktop control*. Tujuan dari validasi adalah untuk mengetahui seberapa besar nilai eror dari respon platform dibandingkan dengan batasan eror yang sudah ditentukan pada metode penelitian.

*“Halaman Ini Sengaja Dikosongkan”*

## BAB IV RANCANG BANGUN MODEL DAN PEMBAHASAN

### 4.1. Gambaran Umum

Pada tahap ini akan dilakukan proses perancangan sistem kendali *Dynamic Positioning System* pada *Platform Semi-Submersible* dengan mengacu pada lima skenario pengujian terhadap 3 DOF (*Degree Of Freedom*). Oleh karena itu pada bab ini akan dibahas lebih mendalam mengenai proses perancangan dan pembuatan Model *Platform Semi-Submersible* serta perancangan sistem kendali *Dynamic Positioning System* pada platform dengan menggunakan *logika Fuzzy*. Setelah dilakukan proses perancangan sistem kendali DPS, Model kemudian akan dilakukan pengujian berdasarkan 5 skenario uji coba yang sudah dirancang terhadap variasi arus yang sudah ditentukan.

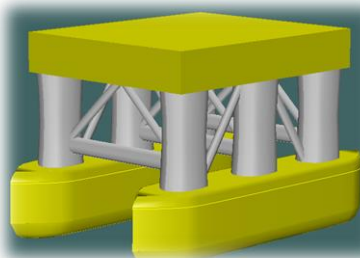
Dalam proses pengujian akan dilakukan dengan skenario penulis tanpa dilakukan proses penyekalaan terhadap kondisi lingkungan platform yang sebenarnya dikarinakan keterbatasan sarana atau tempat pengujian dan hanya terfokus pada arus laut saja sebagai gaya lingkungan yang berdampak.

### 4.2. Perancangan Model Platform Semi-Submersible.

Perancangan Model Platform Semi-submersible dengan *Dynamic Positioning Sistem* terdiri dari perancangan sistem mekanis, dan perancangan sistem elektronik agar platform dapat menjalankan fungsinya untuk *Dynamic Positioning System*. Proses perancangan mekanik sendiri terdiri dari proses desain Model platform dan proses pembuatan Model menjadi nyata. Sedangkan dalam proses perancangan sistem elektronik meliputi proses perancangan komponen-komponen serta sensor-sensor aa saja yang akan digunakan untuk bisa mendukung *Dynamic Positioning System* pada Model *Platform Semi-submersible*.

#### 4.3.1. Perancangan Sistem Mekanik Model

Perancangan sistem mekanik dimulai dengan melakukan pembuatan desain Model platform *Semi-submersible* dengan menggunakan aplikasi desain *Maxsurf*. Dimensi dari Model platform dibuat berdasarkan skala dari ukuran platform yang sesungguhnya dan umum digunakan. tujuan dalam tahap pendesainan Model selain untuk merancang bentuk Model, juga untuk mengetahui nilai-nilai koefisien serta parameter-parameter pada Model platform seperti pada gambar berikut.



Gambar 4. 1 Desain Model *Platform Semi-submersible*.

	Measurement	Value	Units
1	Displacement	11.84	kg
2	Volume (displaced)	0.012	m <sup>3</sup>
3	Draft Amidships	0.100	m
4	Immersed depth	0.100	m
5	WL Length	0.600	m
6	Beam max extents o	0.413	m
7	Wetted Area	0.341	m <sup>2</sup>
8	Max sect. area	0.023	m <sup>2</sup>
9	Sect. area amidships	0.023	m <sup>2</sup>
10	Waterpl. Area	0.124	m <sup>2</sup>
11	Prismatic coeff. (Cp)	0.860	
12	Block coeff. (Cb)	0.822	
13	Max Sect. area coeff	0.956	
14	Waterpl. area coeff.	0.864	
15	LCB length	0.301	from z
16	LCF length	0.301	from z
17	LCB %	50.168	from z
18	LCF %	50.145	from z
19	Length:Beam ratio	2.501	
20	Beam:Draft ratio	2.399	
21	Length:Vol <sup>0.333</sup> rati	2.632	
22	Precision	Medium	55 stati

Gambar 4. 2 Perhirungan *Hydrostatics*.

Dimensi Model *Platform Semi-submersible* yang di desain adalah sebagai berikut :



Gambar 4. 3 Model *Platform Semi-Submersible*



Tabel 4. 1 Dimensi Model *Platform Semi Submersible*

Dimensi Model Platform Semi-submersible		
Parameter	Ukuran	Satuan
L	60	Cm
B	42	Cm
H	35	Cm
T	10	Cm
B ponton	12	Cm
H ponton	12	Cm
H pilar	17	Cm

Untuk mendukung platform semi-submersible dapat bergerak ke segala arah dengan kemampuan manuver yang baik, platform *semi-submersible* menggunakan sistem propulsi *azimuth*. Model platform *semi submersible* ini dilengkapi dengan empat *motor thruster azimuth* sehingga membuat platform dapat bergerak *Surge, sway, yaw*, serta mampu bergerak secara serong. Berikut desain sistem propulsi azimuth yang diterapkan pada Model platform *Semi-submersible*.

Gambar 4. 4 Implementasi Sistem Propulsi *Azimuth*

#### 4.3.2. Perancangan Sistem Elektronik Model

Sistem elektronik pada Model Platform *Semi-submersible* dengan *Dynamic Positioning System* terdiri dari sistem elektronik pada platform dan sistem elektronik pada base pemancar referensi point yang terdapat di darat. Sistem elektronik pada Model terdiri dari komponen-komponen pendukung untuk menjalankan *Dinamic Positioning System* secara otomatis. Secara umum *Dynamic Position* sistem terdiri dari *kontroller*, sistem sensor, *motor thruster*, dan *sistem power*.

Sistem elektronik *Dynamic Positioning System* dengan metode DGPS dapat dibagi menjadi 2 bagian yaitu sistem elektronik pada Model Platform *Semi-submersible* dan sistem elektronik pada based pemancar data GPS sebagai titik referensi di darat.

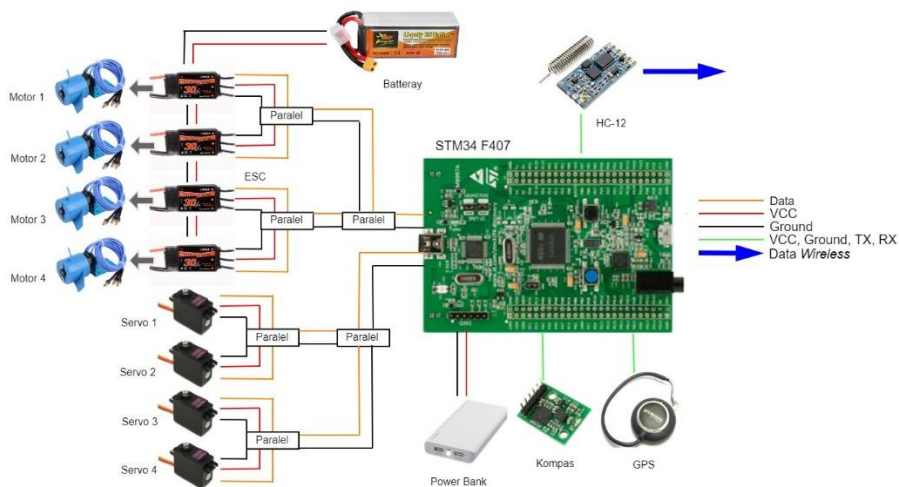
## b. Sistem Elektronik Platform

Sistem elektronik platform terdiri dari kontroler, sensor, aktuator berupa motor thruster dan servo, serta sistem power. Kontroler pada Model ini menggunakan mikrokontroler STM 32. *Mikrokontroler STM 32* dipilih karena memiliki kemampuan proses yang lebih cepat jika dibandingkan dengan mikrokontroler yang lainnya seperti arduino. STM 32 juga memiliki jumlah pin I/O yang lebih banyak. *Mikrokontroler* berfungsi untuk mengatur sistem kerja dari aktuator melalui inputan data yang diterima oleh sensor yang kemudian diolah dengan algoritma yang telah di program.

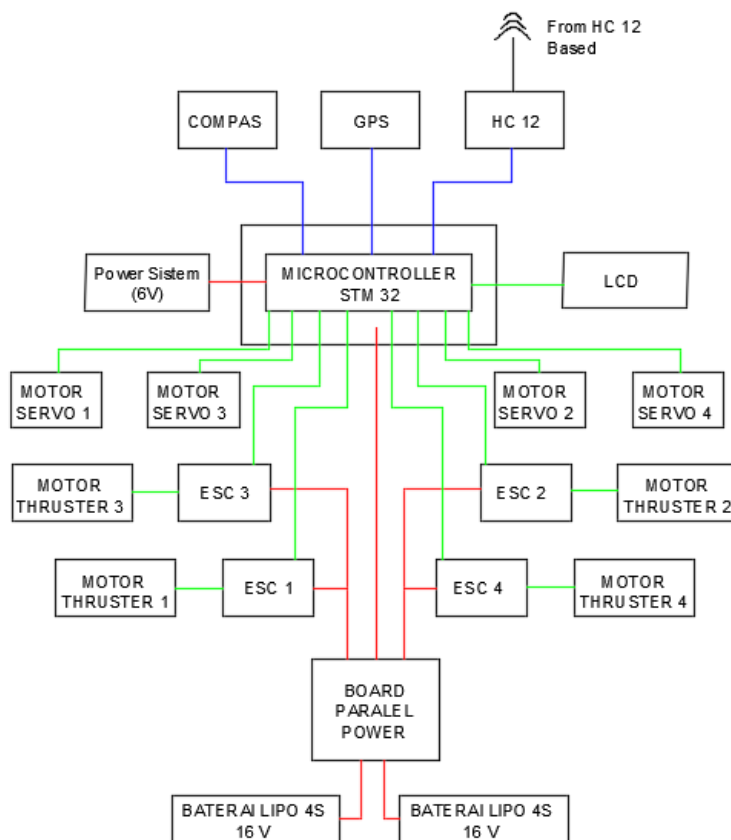
Terdapat 2 sensor yang digunakan, yaitu GPS dan Kompas. Sensor GPS digunakan sebagai inputan untuk menentukan titik referensi tujuan, dan untuk inputan titik posisi platform yang dipertahankan. Data dari sensor GPS berupa nilai titik *Latitude* dan *longitude*. Prinsip kerja sensor GPS untuk bisa menuju ke titik referensi tujuan ataupun untuk mempertahankan posisi adalah dengan mengolah data latitude dan longitude dari 2 titik yaitu posisi platform dan tujuan platform sehingga dengan mencari nilai Delta Y, Delta X, dan sudut dari posisi ke titik tujuan. Sensor Kompas bertujuan untuk memberikan inputan berupa sudut *Heading* Platform yang harus dipertahankan.

*Aktuator* pada Model platform berupa 4 unit azimuth propulsion sistem yang terletak di empat bagian sudut platform. Azimuth propulsion sistem yang diterapkan terdiri dari dua komponen utama yaitu motor thruster sebagai motor penggerak utama pada Model platform dan motor servo untuk mengatur sudut dari motor thruster. Untuk efektifitas dan efisiensi pergerakan sudut motor thruster dibatasi hanya mampu untuk bergerak sampai sudut 90 derajat saja.

Power sistem yang digunakan untuk sistem menggerakkan sistem yang ada menggunakan baterai yang disusun secara paralel untuk memenuhi kebutuhan motor thuster, motor servo, dan sistem. Berikut merupakan gambar rangkaian sistem elektronik pada Model platform.



Gambar 4. 5 *Wiring Diagram* Rangkaian Elektronik pada Model



Rangkaian Elektronik Pada Platform Semi-submersible

Gambar 4. 6 Skema Rangkaian Elektronik Pada *Model Semi-submersible*

Berdasarkan *wiring diagram* diatas, terdapat garis-garis lurus berwarna yang mewakili setiap jenis kabel yang terhubung ke microkotrroller STM32 F407. Pada STM32 F407 terdapat pin yang terhubung ke input dan output. Untuk output pada pin A6, A7, D14, dan D15 adalah untuk output motor Servo. Semetara untuk output B0, B1, D12, D13 adalah untuk output Motor Thruster. Untuk input pin A0 dan Output pin A1 adalah untuk GPS, sedangkan input pin D8 dan Output D9 adalah untuk Kompas, dan untuk Input A2 dan Output A3 adalah untuk HC-12.

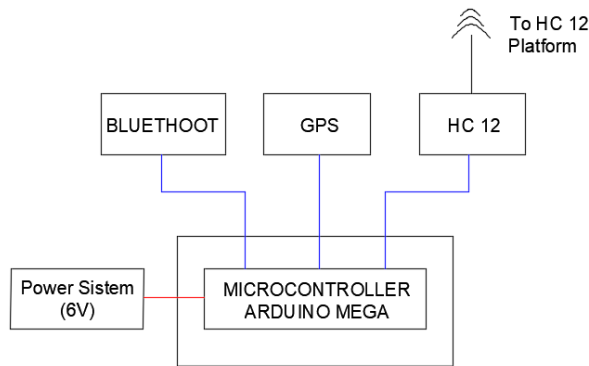
Untuk penjelasan garis-garis pada wiring diagram diatas adalah sebagai berikut. Warna oranye pada diagram diatas adalah kabel data yang berisi perintah yang saling diinformasikan oleh STM32 F407 dan *receiver radio remot control*. Garis merah adalah VCC atau voltase dimana kabel VCC ini berfungsi sebagai aliran tegangan sebagai catu daya pada komponen yang dihubungkan. Garis hitam adalah Ground, dimana garis kabel ini berisi voltase 0 volt dan sebagai minus untuk proteksi peralatan elektronik sehingga dapat mencegah kerusakan akibat daya bocor tegangan. Untuk garis panah biru menggambarkan data

komunikasi yang diterima secara wireless oleh HC-12 dari based ke Model. Dan garis hijau menggambarkan tanda data serial tx rx, artinya menyalurkan datanya melalui komunikasi serial.

c. Sistem Elektronik Based

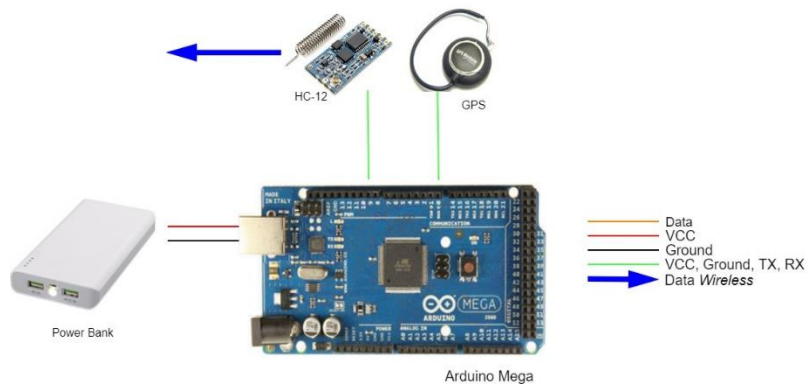
Based merupakan bagian pendukung pada sistem *Dynamic Positioning* dimana based berfungsi untuk menentukan nilai eror GPS di darat sebagai referensi pada GPS yang terdapat di platform. Nilai eror ini didapatkan dari *drifting* yang terjadi pada GPS di based dalam kondisi diam. Nilai eror tersebut kemudian akan dikirimkan ke platform untuk kemudian dikurangkan atau ditambahkan pada data gps yang terapat pada platform sehingga mampu mengurangi drifting dan meningkatkan akurasi dari titik yang dipertahankan.

Komponen penyusun dari based terdiri dari *Mikrokontroler Arduino Mega* yang berfungsi sebagai kontroler utama pada Based untuk melakukan proses pengiriman data GPS ke platform. Sensor yang digunakan adalah GPS untuk memberikan inputan data posisi dari based untuk kemudian dikirim ke Platform melalui HC 12 sebagai komunikasi wireless dengan frekuensi 433 Hz.



Rangkaian Elektronik Pada Based

Gambar 4. 7 Skema Rangkaian Elektronik Pada Based Reference Point



Gambar 4. 8 Wiring Diagram Rangkaian Elektronik pada Based

Dari gambar diagram diatas terdapat gambar garis lurus yang mewakili kabel2 yang terhubung ke secara langsung ke Arduino Mega. Adapun untuk pin input dan output yang digunakan pada microcontroler arduino mega adalah sebagai berikut. Untuk pin input 14 dan output 15 adalah untuk GPS. Sedangkan pin Input 16 dan output 17 adalah untuk HC-12.

Untuk penjelasan warna garis pada pada wiring diagram diatas adalah sebagai berikut. Warna merah adalah VCC atau voltase dimana kabel ini berisi tegangan untuk memberikan catu daya pada komponen yang dihubungkan. Warna hitam adalah kabel ground yang berisi tegangan 0 volt sebagai proteksi peralatan elektronik sehingga dapat mencegah kerusakan akibat daya bocor tegangan dan menetralsisir *noise* yang disebabkan kualitas komponen yang tidak standar. Sedangkan garis hijau adalah tanda data serial tx, rx, yang artinya perangkat menyalurkan datanya melalui komunikasi serial.

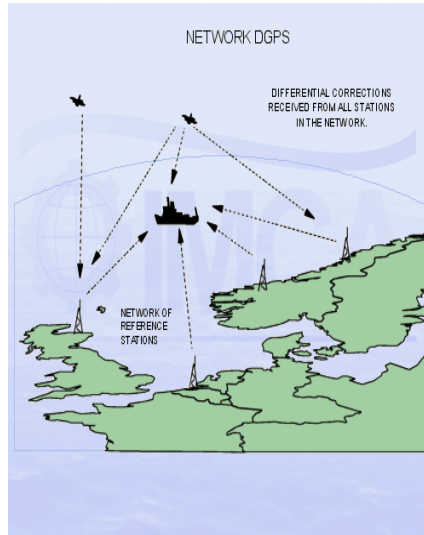
Berikut merupakan gambar Implementasi based sebagai refrensi point platform dan Model *Platform Semi-submersible* dengan sistem DGPS.



Gambar 4. 9 Implementasi Based Reference Point

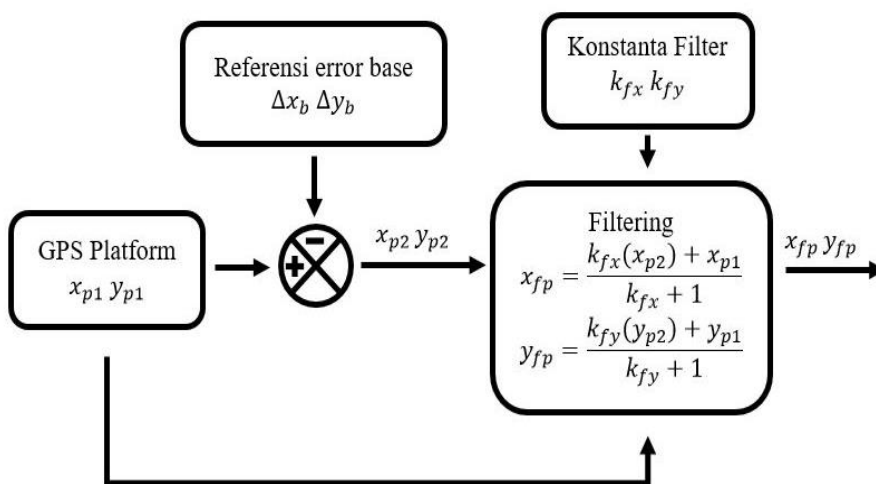
### 4.3. Perancangan Sistem Dynamic Positioning

Sistem Dinamic Positioning terdiri dari beberapa metode utama yang biasa di terapkan pada kapal ataupun bangunan terapung lainnya yaitu meliputi metode *HPR(Hydroacoustic Position Reference)*, *Taut Wire*, *DGPS(Differentian Global Positioning System)*, *Laser Based System*, dan *Artemis*. Pada perancangan sistem Dynamic Positioning yang di terapkan di Platform Semi-submersibe ini menggunakan metode *DGPS (Differential Global Positioning System)*.



Gambar 4. 10 Sistem DPS dengan Metode DGPS

*Differential Global Positioning System* merupakan salah satu metode DPS pada platform dengan memanfaatkan titik refference berupa titik GPS. Dimana pada kondisi kapal yang sebenarnya, nilai eror object terhadap titik refrensi GPS mencapai 20 m, sedangkan pada modul GPS Ublox M8N yang digunakan pada Model memiliki nilai eror 1.5m. Untuk memperkecil nilai eror tersebut ditambahkan lah titik refrensi baru pada based yang terdapat di darat. Based yang terdapat didarat berfungsi untuk mendapatkan nilai eror dari gps yang mengalami drifting. Nilai eror tersebutlah yang kemudian akan dikirimkan ke sistem kontrol platform yang kemudian akan ditambahkan atau dikurangkan pada GPS yang terdapat di platform untuk meningkatkan akurasi titik di platform. Berikut merupakan rancangan algoritma metoda DGPS.



Gambar 4. 11 Rancangan Filter DGPS untuk mengurangi Error GPS

$$\Delta x_b = x_{b1} - x_{b0}$$

$$\Delta y_b = y_{b1} - y_{b0}$$

$$x_{p2} = x_{p1} + \Delta x_b$$

$$y_{p2} = y_{p1} + \Delta y_b$$

$$x_{fp} = \frac{k_{fx}(x_{p2}) + x_{p1}}{k_{fx} + 1}$$

$$y_{fp} = \frac{k_{fy}(y_{p2}) + y_{p1}}{k_{fy} + 1}$$

dimana,

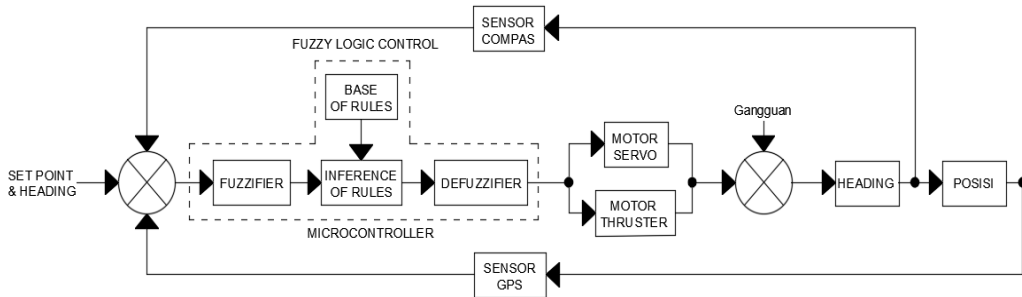
- $x_{b0}$  = posisi *longitude fixed base* yang sudah diketahui sebelumnya
- $x_{b1}$  = posisi *longitude fixed base* berdasarkan *input GPS* pada *base*
- $y_{b0}$  = posisi *latitude fixed base* yang sudah diketahui sebelumnya
- $y_{b1}$  = posisi *latitude fixed base* berdasarkan *input GPS* pada *base*
- $x_{p1}$  = posisi *longitude platform* berdasarkan *input GPS* pada *platform*
- $x_{p2}$  = hasil penjumlahan *longitude platform* dengan *error base*
- $y_{p1}$  = posisi *latitude platform* berdasarkan *input GPS* pada *platform*
- $y_{p2}$  = hasil penjumlahan *latitude platform* dengan *error base*
- $k_{fx}$  = konstanta filter *longitude*
- $k_{fy}$  = konstanta filter *latitude*
- $x_{fp}$  = posisi *longitude platform* setelah difilter
- $y_{fp}$  = posisi *latitude platform* setelah difilter

Hasil dari filter  $x_{fp}$  dan  $y_{fp}$  akan didefinisikan sebagai  $xP$  dan  $yP$  atau posisi platform saat ini yang diharapkan memiliki nilai eror kurang dari 0.5 m. Untuk nilai konstanta filter  $k_{fx}$  dan  $k_{fy}$  akan dilakukan percobaan untuk mendapatkan nilai yang paling optimal pada rentang konstanta filter 1 - 20.

#### 4.4. Diagram Blok Kontrol Sistem

Perancangan sistem kontrol DPS pada Model *Semi-submersible Platform* pertama kali dimulai dengan pembuatan diagram blok alur sistem kontrol. Diagram block sistem kontrol bertujuan untuk mengetahui alur proses kerja sistem kontrol yang diterapkan pada Model platform semisubmersible. Sehingga dengan adanya blok diagram dapat mempermudah proses perancangan sistem kontrol pada tahap selanjutnya menggunakan *Fuzzy Logic Control*.

Blok diagram yang digunakan pada Rancang Bangun *Dynamic Positioning System* pada Model *Semi-submersible Platform* dibuat sebagaimana mestinya seperti dengan sistem kontrol otomatis pada umumnya. Karena sistem kontrol dynamic position merupakan jenis sistem kontrol yang berulang secara otomatis untuk selalu bisa mempertahankan posisi dan arah heading platform, diagram blok yang dibuat merupakan diagram blok dengan sistem *Close Loop*. Berikut diagram blok sistem kontrol pada Model platform Semi Submersible.



Gambar 4. 12 Diagram Block Dynamic Positioning System

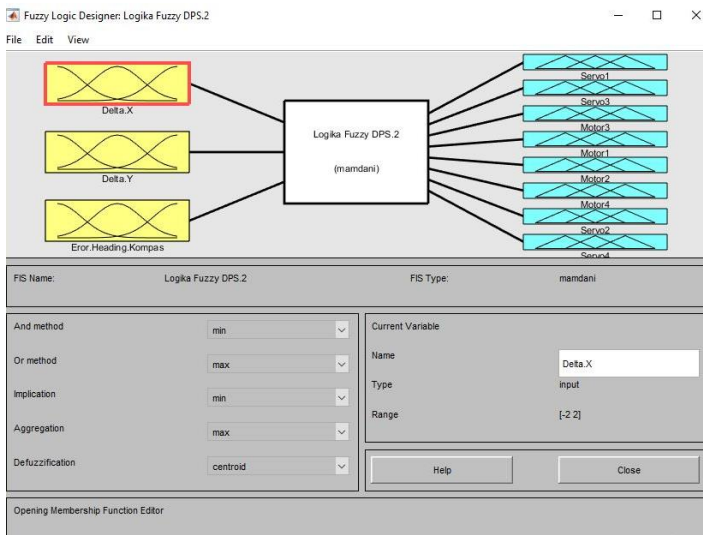
Pada gambar diagram block diatas dapat kita ketahui bahwa setelah set point ditentukan untuk mengkontrol titik referensi dan heading platform, langkah pertama yang dilakukan adalah memproses sistem kerja DPS secara otomatis melalui *microcontroller*. Di dalam mikrokontroller sudah diisi oleh algoritma program untuk memproses data input sehingga dapat mengaktifkan aktuator yang dimana disini terdapat 2 aktuator yaitu servo dan motor thruster. Dimana servo digunakan untuk mengatur sudut motor sesuai dengan arah perpindahan platform dan arah heading terhadap posisi platform. Dan motor thruster sebagai penggerak utama pada platform menyesuaikan kecepatan motor terhadap jarak perpindahannya.

Untuk mengetahui posisi tujuan dan eror posisi platform terhadap tujuan digunakanlah sensor GPS. Sensor GPS tersebut akan memberikan input berupa posisi platform dan posisi tujuan yang kemudian didapatkan nilai inputan delta Y dan delta X sebagai inputan. Selain itu, untuk mengatur agar *heading* platform tetap terjaga digunakan sensor kompas yang akan memberikan input nilai eror *heading* platform terhadap heading tujuan platform.

#### 4.5. Perancangan Sistem Kontrol Fuzzy

*Fuzzy Logic Control* merupakan salah satu jenis sistem kontrol yang biasa digunakan karena *Fuzzy Logic Control* dapat digunakan untuk mampu menyelesaikan sebuah sistem kontrol yang kompleks. Pada perancangan sistem kontrol DPS Model platform Semi-submersible ini menggunakan metode logika *Fuzzy Mamdani*. Metode Mamdani sering dikenal sebagai metode Max-Min dimana untuk mendapatkan output-nya dibutuhkan 4 tahap yaitu, pembentukan himpunan Fuzzy, aplikasi fungsi implikasi, dan komposisi aturan. Berikut merupakan desain logika fuzzy sistem Kontrol DPS pada Model Semi-submersible Platform.





Gambar 4. 13 Desain Kontrol Fuzzy DPS

Pada gambar diatas menunjukkan desain rancangan kontrol fuzzy melalui Matlab. Pada desain kontrol Fuzzy tersebut terdapat input, rules, serta output. Dalam sistem DPS ini input yang dibutuhkan oleh Model platform untuk bisa mempertahankan posisi dan heading adalah Delta X, Delta Y yang didapatkan dari pengolahan data kompas serta nilai iputan eror terhadap heading platform yang di dapatkan dari pengolahan data kompas. Untuk output pada platform sendiri terdiri dari 4 motor trhuster dan 4 motor servo.

Pada desain logika fuzzy diatas menggunakan metoda fuzzy mamdani atau metoda max-min dimana untuk nilai input logika AND maka nilai yang diambil adalah min dan untuk logika OR nilai yang diambil adalah Max. Dan menggunakan fungsi segitiga untuk pengelompokan anggota fungsi himpunan fuzzy. Metoda Mamdani memberikan output berupa nilai yang sesungguhnya tidak berupa bilangan linguistik. Untuk metoda defuzzifikasi menggunakan metoda *centroid* dimana nilai output akan diambil berdasarkan luasan atau nilai tengah dari setiap kondisi input yang berpengaruh pada nilai output.

#### 4.5.1. *Membership Function Input*

Dalam penentuan nilai input sistem delta x dan delta y tidak ditentukan secara langsung dari data yang diterima oleh kompas berupa data latitude dan longitude yang sifatnya konstan terhadap bumi. Nilai eror yang dibutuhkan untuk platform adalah nilai eror delta x dan delta y terhadap arah heading Model. Untuk mendapatkan nilai eror tersebut, maka inputan data latitude dan longitude dari kompas harus dirotasi terlebih dahulu sehingga bisa mendapatkan persamaan nilai yang baru.

Berikut merupakan persamaan untuk mengkonversikan nilai eror perpindahan longitude dan latitude menjadi jarak dalam satuan meter.

$$\Delta x = (xP - xT) \times Ln$$

dimana,

$\Delta x$  = jarak membujur antara *platform* dengan titik yang dipertahankan

$xP$  = *longitude platform* dalam desimal, hasil filter

$xT$  = *longitude tujuan* dalam decimal

$Ln$  = konstanta *longitude* desimal ke jarak meter = 110574,6108

$$\Delta y = (yP - yT) \times Lt$$

dimana,

$\Delta y$  = jarak melintang antara *platform* dengan titik yang dipertahankan

$yP$  = *latitude platform* dalam desimal, hasil filter

$yT$  = *latitude tujuan* dalam desimal, dari GPS pada *fixed base*

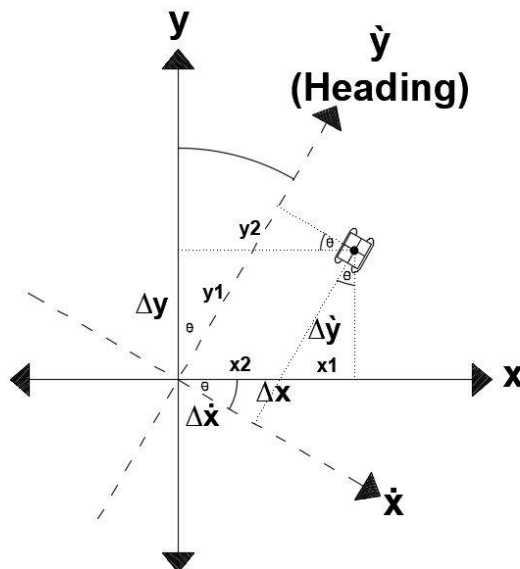
$Lt$  = konstanta *latitude* desimal ke jarak meter = 111320

$$r = \sqrt{|\Delta y^2 - \Delta x^2|}$$

dimana,

$r$  = jarak antara platform dengan titik tujuan

Berikut merupakan skema penentuan nilai eror delta x dan delta y yang baru pada sudut heading yang sudah ditentukan.



Gambar 4. 14 Skema penentuan nilai eror delta x dan delta y

Berikut merupakan penurunan persamaan dari data kompas untuk kemudian bisa mendapatkan nilai eror dari platform berupa nilai eror delta x dan delta y terhadap arah heading platform.

Penurunan persamaan untuk mendapatkan nilai eror delta x :

$$\Delta x = x1 + x2$$

$$x2 = \Delta x - x1$$

$$x1 = \Delta y \times \tan \theta$$

$$x2 = \Delta x - (\Delta y \times \tan \theta)$$

$$\Delta x' = x2 \times \cos \theta$$

$$\Delta x' = (\Delta x - (\Delta y \times \tan \theta)) \cos \theta$$

$$\Delta x' = (\Delta x \times \cos \theta) - (\Delta y \times \sin \theta)$$

Penurunan persamaan untuk mendapatkan nilai eror delta y :

$$y1 = \frac{\Delta y}{\cos \theta}$$

$$y2 = \Delta x' \times \tan \theta$$

$$\Delta y' = y1 + y2$$

$$\Delta y' = \frac{\Delta y}{\cos \theta} + \Delta x' \times \tan \theta$$

dimana,

$\Delta x$  = error jarak membujur

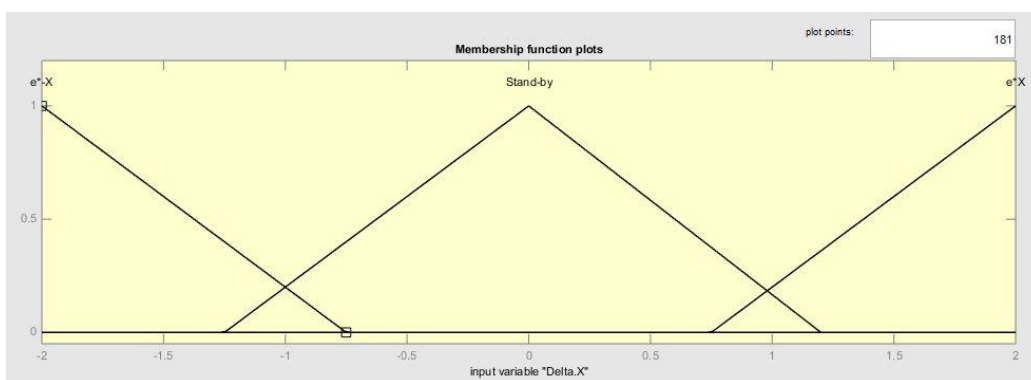
$\Delta y$  = error jarak melintang

$\theta$  = sudut yang dipertahankan *platform*

$\Delta x'$  = error jarak membujur setelah dipengaruhi  $\theta$

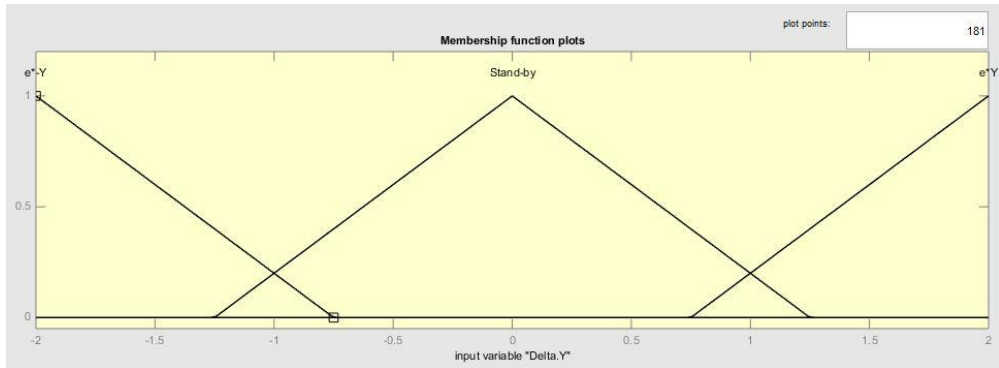
$\Delta y'$  = error jarak melintang setelah dipengaruhi  $\theta$

Dari persamaan diatas dapat ditentukan nilai eror delta X dan delta Y yang baru terhadap arah heading kapal pada sudut tertentu dengan mengolah data latitude dan longitude yang didapatkan dari data GPS. Nilai eror delta X dan Delta Y tersebut kemudian dikelompokkan berdasarkan kondisi eror sesuai dengan rules yang ditentukan sebagai berikut.



Gambar 4. 15 Merbership Function Input Delta X

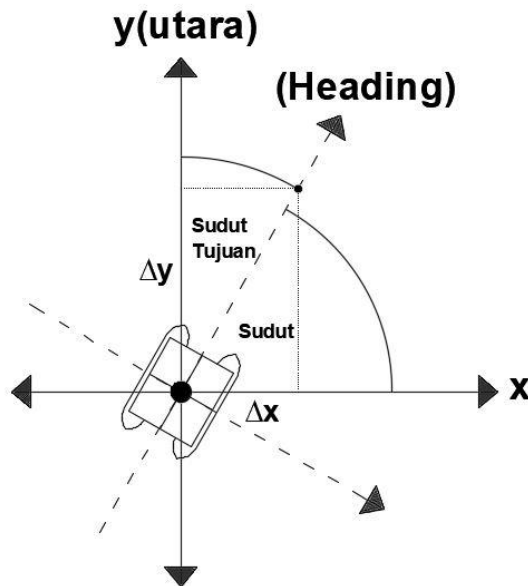
Pada nilai inputan membership function untuk kondisi DPS dibedakan menjadi 3 kondisi yaitu stand-by dimana nilai inputan delta x berkisar dari 1m sampai -1 m, kemudian e\*X yaitu dari 1m sampai 2m, dan e\*-X yaitu dari -1m sampai -2m.



Gambar 4. 16 Membership Function Input Delta Y

Pada nilai inputan membership function DPS pada inputan Delta Y sama seperti pada membership function Delta X dimana terdapat 3 kondisi fungsi yaitu Stand-by (1m sampai -1m),  $e*X$  (1m sampai 2m) dan  $e*-X$  (-1m sampai -2m).

Nilai input yang ketiga adalah nilai input sudut eror terhadap heading atau sudut tujuan platform. Dimana untuk mendapatkan nilai eror sudut terhadap heading platform menggunakan nilai input dari GPS untuk menentukan sudut terhadap kuadran satu terlebih dahulu, kemudian setelah itu merubah sudut dari kuadran satu menjadi nilai sudut sesuai dengan kondisi kompas dimana nilai 0 derajat berada pada sumbu y. Setelah itu di dapatkanlah sudut tujuan atau heading platform, untuk mencari eror sudut maka nilai sudut heading dikurangi dengan nilai sudut dari sensor kompas. Berikut merupakan alur untuk mendapatkan nilai eror sudut heading.



Gambar 4. 17 Skema Penentuan Error Heading

Perhitungan sudut tujuan terhadap sumbu x (kuadran),

$$rad = sudut \times \left(\frac{\pi}{180}\right) \qquad rad = Atan \times (sudut)$$

$$sudut = rad \times \left(\frac{180}{\pi}\right) \qquad rad = Atan2 (\Delta y, \Delta x)$$

$$sudut = Atan 2 (\Delta y, \Delta x) \times \left(\frac{180}{\pi}\right)$$

Perhitungan sudut tujuan terhadap sumbu y (sudut sebenarnya),

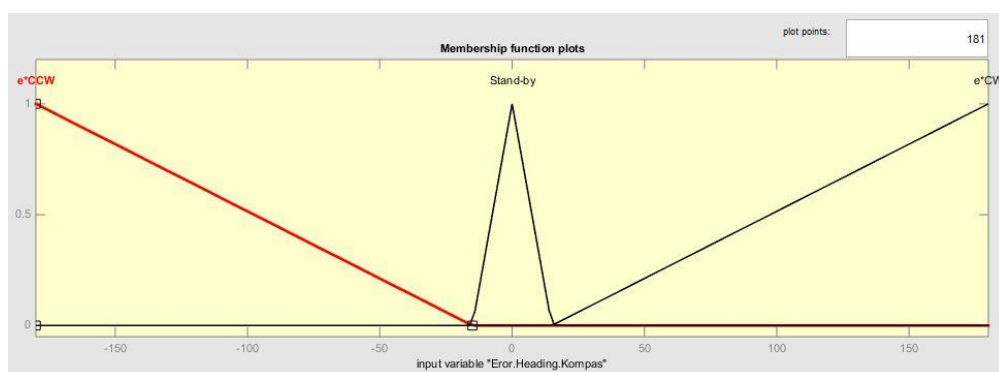
$$sudut\ tujuan = mod((450 - sudut), 360)$$

Perhitungan eror sudut tujuan (heading) terhadap sudut kompas,

$$eror\ sudut\ navigasi = (sudut\ tujuan - sudut\ kompas)$$

$$eror\ sudut = (sudut\ kompas - sudut\ heading)$$

Setelah di dapat nilai eror sudut, kemudian dilakukan pengelompokkan nilai eror sesuai dengan kondisi dibawah.



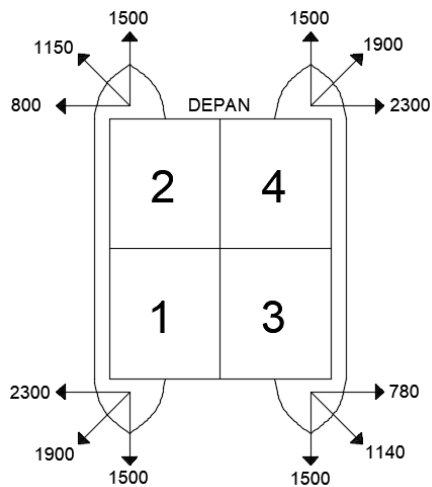
Gambar 4. 18 Membership Function Error Heading

Nilai input eror heading dibedakan menjadi 3 membership function juga, yaitu kondisi Stand-by dimana nilai eror yang dijadikan inputan berkisar dari -15 sampai 15 derajat, kemudian e\*CW yaitu nilai inputannya berkisar dari 15 sampai 180 derajat dan e\*CCW yaitu nilai inputannya berkisar dari -15 sampai -180 derajat.

#### 4.5.2. Membership Function Output

Output pada Model *Semi-submersible Platform* berupa 4 motor *thruster* sebagai penggerak platform dan 4 motor servo sebagai pengatur sudut pergerakan motor *thruster*. Untuk bisa mengkondisikan kecepatan motor thruster dan pergerakan motor servo diperlukan data dari kecepatan motor dan pergerakan servo. Untuk mengetahui data pergerakan motor tersebut dilakukan percobaan pergerakan manual terlebih dahulu sehingga bisa di dapatkan nilai kecepatan dan pergerakan dalam bentuk rentang data

PWM (*Pulse Width Modulation*) yaitu mengubah data pulsa menjadi data frekuensi yang tetap. Berikut merupakan gambaran nilai pergerakan motor servo dalam PWM.

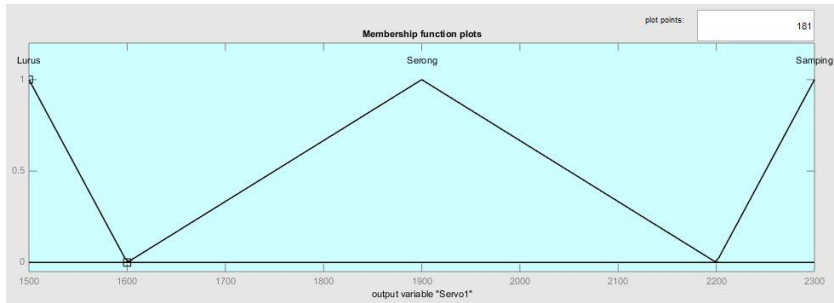


Gambar 4. 19 Data PWM Pergerakan Servo

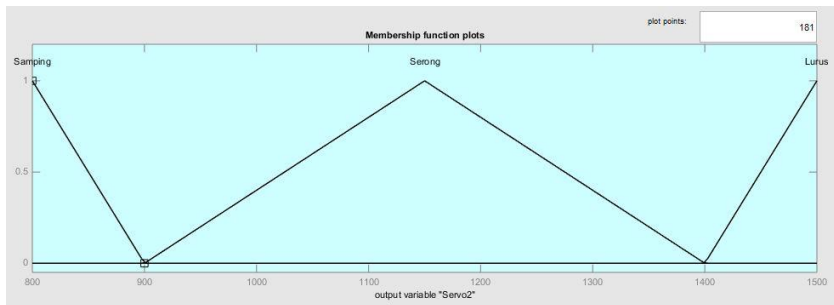
Tabel 4. 2 Data PWM Pergerakan Servo Terhadap Sudut

Data PWM motor Servo terhadap Sudut		
Servo	PWM	Sudut
1	2300	90
	1900	45
	1500	0
2	800	90
	1150	45
	1500	0
3	780	90
	1140	45
	1500	0
4	2300	90
	1900	45
	1500	0

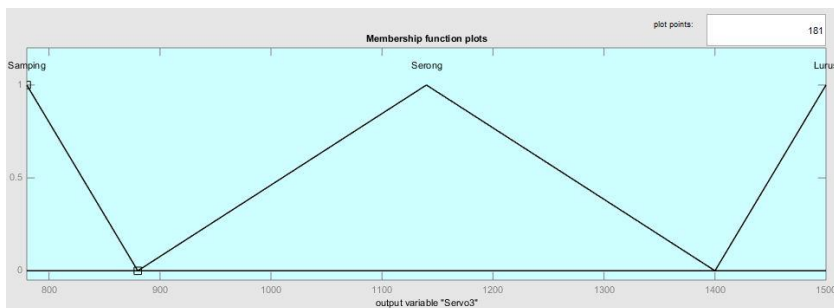
Gambar diatas menunjukkan data PWM pergerakan motor servo pada 3 kondisi. Masing-masing servo memiliki data pergerakan yang berbeda karena tergantung kondisi mekanis sistem propulsi. Dari data masing-masing servo tersebut kemudian kita kelompokkan membership functionnya menjadi 3 mode pergerakan di setiap servo yaitu Lurus, Serong dan Samping. Berikut merupakan pengelompokan data pergerakan di masing-masing servo.



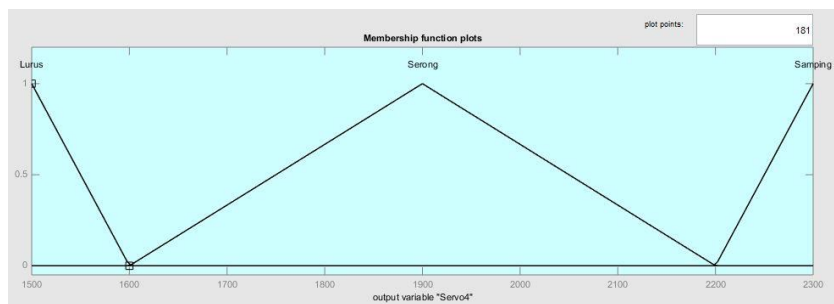
Gambar 4. 20 Membership Fuction Servo 1



Gambar 4. 21 Membership Function Servo 2



Gambar 4. 22 Membership Function Servo 3



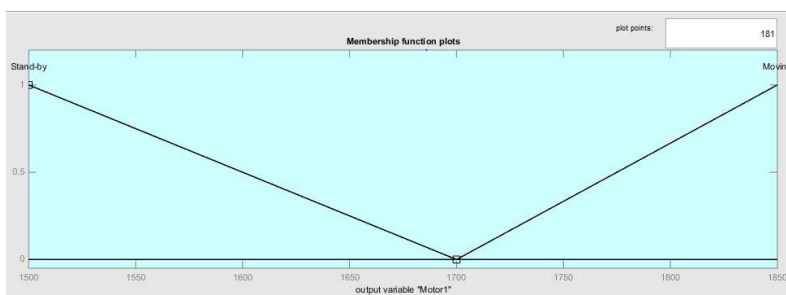
Gambar 4. 23 Membership Function Servo 4

Untuk data PWM pada pergerakan motor Thruster dikelompokkan berdasarkan kecepatannya. Untuk data percobaan manual didapatkan bahwa nilai PWM kecepatan motor berkisar dari 1500 sampai 1850. Dimana 1500 merupakan kondisi dimana motor tidak bergerak, dan akan mulai bergerak pada saat data PWM motor mencapai 1550. Dan 1850 merupakan kondisi motor berputar dengan kecepatan Maksimal.

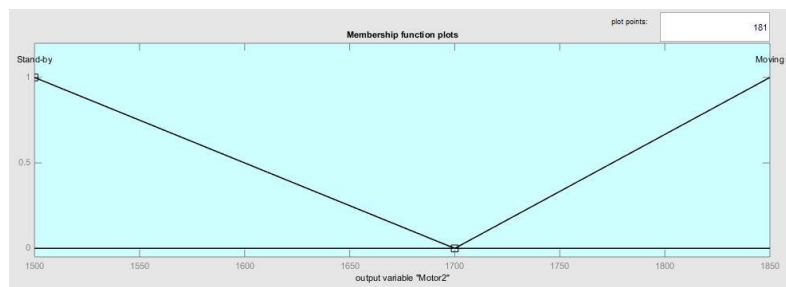
Tabel 4. 3 Data PWM *Thruster* terhadap Kecepatan dan *Thrust*

Data PWM Motor Thruster terhadap Power dan Kecepatan, (Data Spec. Motor)		
PWM	Kecepatan (RPM)	Thrust (Kg)
1500	0	0
1550	743	0.198
1600	1486	0.396
1650	2229	0.595
1700	2971	0.793
1750	3714	0.991
1800	4457	1.189
1850	5200	1.388

Membership Function pada motor thruster dikelompokkan menjadi 2 kondisi yaitu ketika motor bergerak dalam kondisi *Stand-by* dan ketika motor bergerak dalam kondisi *Moving*. Berikut merupakan *membership Function* pada setiap motor *thruster*.

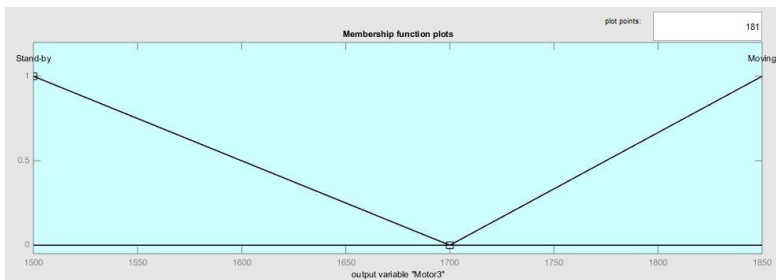


Gambar 4. 24 *Membership Function* Motor 1

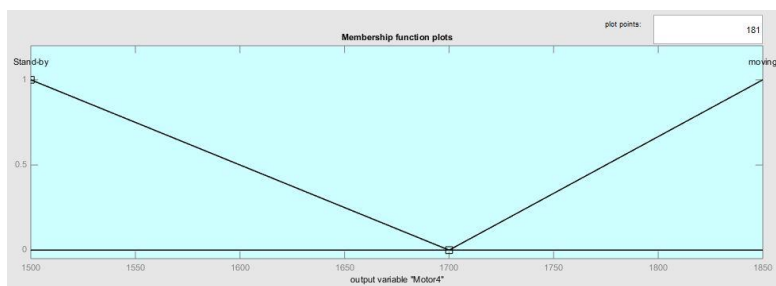


Gambar 4. 25 *Membership Function* Motor 2





Gambar 4. 26 Membership Function Motor 3



Gambar 4. 27 Membership Function Motor 4

### 4.5.3. Rule Based Fuzzy

*Rule Based Fuzzy* merupakan aturan yang diterapkan pada sistem kontrol Fuzzy. Aturan ini menjadi dasar kontrol untuk setiap kondisi yang ada berdasarkan algoritmanya. Dari membership function yang sudah dibuat di Input dan di Output tersebut kebudian dibikin aturan utamanya agar nilai input yang diterima akan mendapatkan respon output yang sesuai. Dengan menggunakan matlab berikut merupakan rule based yang dibuat pada sistem kontrol DPS Model *Semi-submersible Platform*.

```

1. If (Delta X is Stand-by) and (Delta Y is Stand-by) and (Error Heading Kompas is Stand-by) then (Servo1 is Serong)(Servo3 is Serong)(Motor3 is Stand-by)(Motor1 is Moving)(Motor2 is Stand-by)(Motor4 is Stand-by)(Servo2 is Serong)(Servo4 is Serong) (1)
2. If (Delta X is Stand-by) and (Delta Y is e^-Y) and (Error Heading Kompas is Stand-by) then (Servo1 is Lurus)(Servo3 is Lurus)(Motor3 is Moving)(Motor1 is Moving)(Motor2 is Stand-by)(Motor4 is Stand-by)(Servo2 is Serong)(Servo4 is Serong) (1)
3. If (Delta X is Stand-by) and (Delta Y is e^Y) and (Error Heading Kompas is Stand-by) then (Servo1 is Serong)(Servo3 is Serong)(Motor3 is Stand-by)(Motor1 is Stand-by)(Motor2 is Moving)(Motor4 is moving)(Servo2 is Lurus)(Servo4 is Lurus) (1)
4. If (Delta X is e^-X) and (Delta Y is Stand-by) and (Error Heading Kompas is Stand-by) then (Servo1 is Serong)(Servo3 is Lurus)(Motor3 is Stand-by)(Motor1 is Moving)(Motor2 is Moving)(Motor4 is Stand-by)(Servo2 is Serong)(Servo4 is Lurus) (1)
5. If (Delta X is e^X) and (Delta Y is Stand-by) and (Error Heading Kompas is Stand-by) then (Servo1 is Lurus)(Servo3 is Serong)(Motor3 is Moving)(Motor1 is Stand-by)(Motor2 is Stand-by)(Motor4 is moving)(Servo2 is Lurus)(Servo4 is Serong) (1)
6. If (Delta X is e^-X) and (Delta Y is e^-Y) and (Error Heading Kompas is Stand-by) then (Servo1 is Serong)(Servo3 is Lurus)(Motor3 is Stand-by)(Motor1 is Moving)(Motor2 is Moving)(Motor4 is Stand-by)(Servo2 is Serong)(Servo4 is Serong) (1)
7. If (Delta X is e^-X) and (Delta Y is e^Y) and (Error Heading Kompas is Stand-by) then (Servo1 is Serong)(Servo3 is Serong)(Motor3 is Stand-by)(Motor1 is Moving)(Motor2 is Moving)(Motor4 is Stand-by)(Servo2 is Serong)(Servo4 is Lurus) (1)
8. If (Delta X is e^X) and (Delta Y is e^-Y) and (Error Heading Kompas is Stand-by) then (Servo1 is Lurus)(Servo3 is Serong)(Motor3 is Moving)(Motor1 is Stand-by)(Motor2 is Stand-by)(Motor4 is moving)(Servo2 is Serong)(Servo4 is Serong) (1)
9. If (Delta X is e^X) and (Delta Y is e^Y) and (Error Heading Kompas is Stand-by) then (Servo1 is Serong)(Servo3 is Serong)(Motor3 is Moving)(Motor1 is Stand-by)(Motor2 is Moving)(Motor4 is moving)(Servo2 is Lurus)(Servo4 is Serong) (1)
10. If (Delta X is Stand-by) and (Delta Y is Stand-by) and (Error Heading Kompas is e^CW) then (Servo1 is Lurus)(Servo3 is Serong)(Motor3 is Moving)(Motor1 is Stand-by)(Motor2 is Moving)(Motor4 is Stand-by)(Servo2 is Serong)(Servo4 is Lurus) (1)
11. If (Delta X is Stand-by) and (Delta Y is Stand-by) and (Error Heading Kompas is e^CW) then (Servo1 is Serong)(Servo3 is Lurus)(Motor3 is Stand-by)(Motor1 is Moving)(Motor2 is Stand-by)(Motor4 is moving)(Servo2 is Lurus)(Servo4 is Serong) (1)
    
```

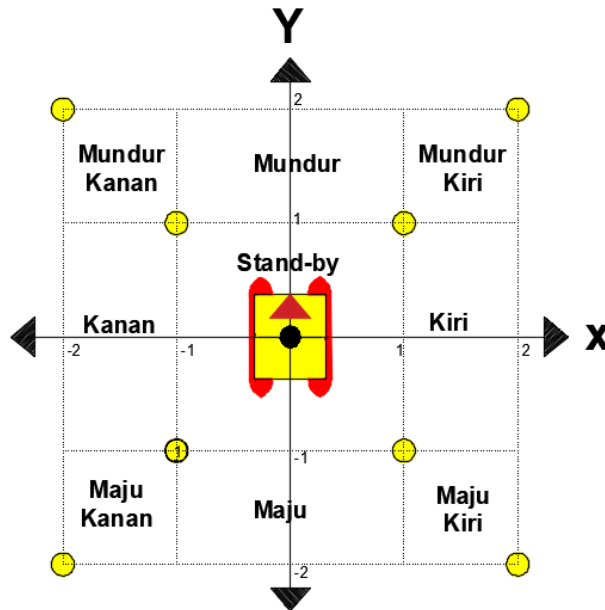
Gambar 4. 28 Rule Based DPS Kontrol

Terdapat 11 Rule Based seperti yang tertera di gambar diatas adalah sebagai berikut,

1. If (Delta X is Stand-by) and (Delta Y is Stand-by) and (Error.Heading.Kompas is Stand-by) Then (Servo 1 is Serong) (Servo 3 is serong) (Motor 3 is Stand-by) (Motor 1 is Stand-by) (Motor 2 is Stand-by) (Motor 4 is Stand-by) (Servo 2 is Serong) (Servo 4 is Serong)

2. If (Delta X is Stand-by) and (Delta Y is e\*-Y) and (Error.Heading.Kompas is Stand-by) then (Servo 1 is Lurus ) (Servo 3 is Lurus ) (Motor 3 is Moving ) (Motor 1 is Moving ) (Motor 2 is Stand-by ) (Motor 4 is Stand-by ) (Servo 2 is Samping) (Servo 4 is Samping)
3. If ( Delta X is Stand-by) and (Delta Y is e\*Y) and (Error.Heading.Kompas is Stand-by) then (Servo 1 is Samping) (Servo 3 is Samping) (Motor 3 is Stand-by) (Motor 1 is Stand-by) (Motor 2 is Moving) (Motor 4 is Moving) (Servo 2 is Lurus) (Servo 4 is Lurus)
4. If (Delta X is e\*-X) and (Delta Y is Stand-by) and (Error.Heading.Kompas is Stand-by) then (Servo 1 is Samping) (Servo 3 is Lurus) (Motor 3 is Stand-by) (Motor 1 is Moving) (Motor 2 is Moving) (Motor 4 is Stand-by) (Servo 2 is Samping) (Servo 4 is Lurus)
5. If (Delta X is e\*X) and (Delta Y is Stand-by) and (Error.Heading.Kompas is Stand-by) then (Servo 1 is Lurus) (Servo 3 is Samping) (Motor 3 is Moving) (Motor 1 is Stand-by) (Motor 2 is Stand-by) (Motor 4 is Moving) (Servo 2 is Lurus) (Servo 4 is Samping)
6. If (Delta X is e\*-X) and (Delta Y is e\*-Y) and (Error.Heading.Kompas is Stand-by) then (Servo 1 is Serong) (Servo 3 is Lurus) (Motor 3 is Stand-by) (Motor 1 is Moving) (Motor 2 is Moving) (Motor 4 is Stand-by) (Servo 2 is Samping) (Servo 4 is Samping)
7. If (Delta X is e\*-X) and (Delta Y is e\*Y) and (Error.Heading.Kompos is Stand-by) then (Servo 1 is Samping) (Servo 3 is Samping) (Motor 3 is Stand-by) (Motor 1 is Moving) (Motor 2 is Moving) (Motor 4 is Stand-by) (Servo 2 is Serong) (Servo 4 is Lurus)
8. If (Delta X is e\*X) and (Delta Y is e\*-Y) and (Error.Heading.Kompos is Stand-by) then (Servo 1 is Lurus) (Servo 3 is Serong) (Motor 3 is Moving) (Motor 1 is Stand-by) (Motor 2 is Stand-by) (Motor 4 is Moving) (Servo 2 is Samping) (Servo 4 is Samping)
9. If (Delta X is e\*X) and (Delta Y is e\*Y) and (Error.Heading.Kompos is Stand-by) then (Servo 1 is Samping) (Servo 3 is Samping) (Motor 3 is Moving) (Motor 1 is Stand-by) (Motor 2 is Stand-by) (Motor 4 is Moving) (Servo 2 is Lurus) (Servo 4 is Serong)
10. If (Delta X is Stand-by) and (Delta Y is Stand-by) and (Error.Heading.Kompos is e\*CCW) then (Servo 1 is Lurus) (Servo 3 is Samping) (Motor 3 is Moving) (Motor 1 is Stand-by) (Motor 2 is Moving) (Motor 4 is Stand-by) (Servo 2 is Samping) (Servo 4 is Lurus)
11. If (Delta X is Stand-by) and (Delta Y is Stand-by) and (Error.Heading.Kompos is e\*CW) then (Servo 1 is Samping) (Servo 3 is Lurus ) (Motor is Sytand-by) (Motor 1 is Moving) (Motor 2 is Stand-by) (Motor 4 is Moving) (Servo 2 is Lurus) (Servo 4 is Samping)

#### 4.6. Respon Pergerakan

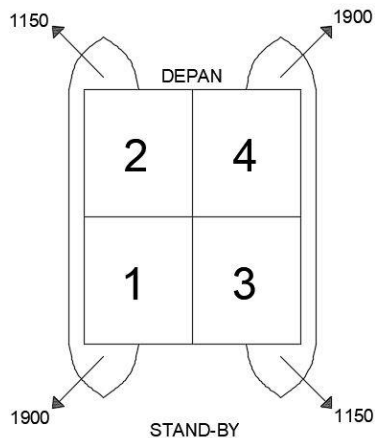


Gambar 4. 29 Skema Pergerakan Model

Berdasarkan *fuzzy rules* yang sudah dibuat dengan menggunakan *Mathlab*, respon pergerakan pada Model *Semi-submersible platform* dibagi menjadi 5 respon pergerakan. Yaitu saat kondisi *Stand-by*, *Surge*, *Sway*, *Yaw*, kombinasi pergerakan *Surge* dan *Sway* (Serong).

##### 4.6.1. *Stand-by*

Respon pergerakan *Stand-by* merupakan respon pergerakan saat Model berada di area rentang eror terhadap *Set point* dengan radius 1 m, yaitu ketika nilai delta x, delta y, dan eror Heading kompas pada *membership function Stand-by*.

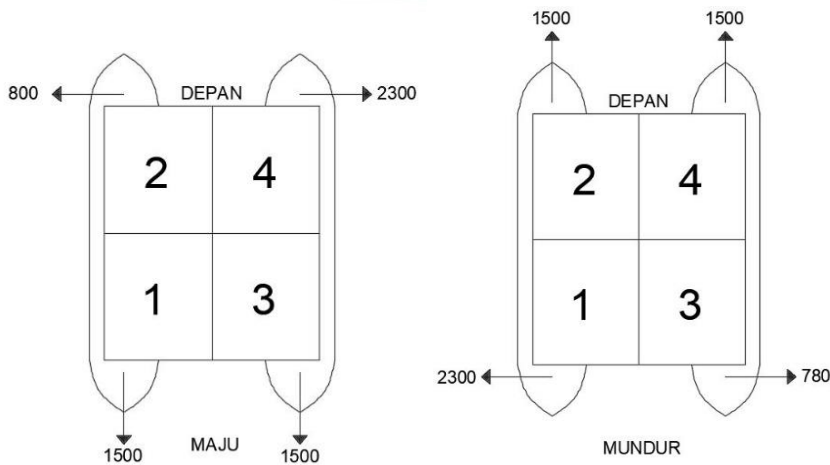


Gambar 4. 30 Respon posisi arah hadap motor kondisi *Stand-by*

Dalam kondisi stand by posisi sudut motor akan mengarah ke posisi serong atau 45 derajat. Dan kecepatan motor thruster berada pada kondisi *membership function stand-by* yaitu antara 1500-1700 pada rentang nilai PWM. Kecepatan motor akan berputar menyesuaikan kondisi eror, semakin besar nilai eror terhadap set point kecepatan motor akan semakin tinggi mendekati 1700 PWM.

#### 4.6.2. Surge

Respon pergerakan *Surge* adalah respon pergerakan platform terhadap kondisi arus air yang memberikan gaya pada bagian haluan atau belakang platform. Respon pergerakan platform Surge ini terdiri dari pergerakan maju dan mundur menuju set point ketika nilai rentang eror delta y berada pada jarak 1m sampai 2 m dari set point.

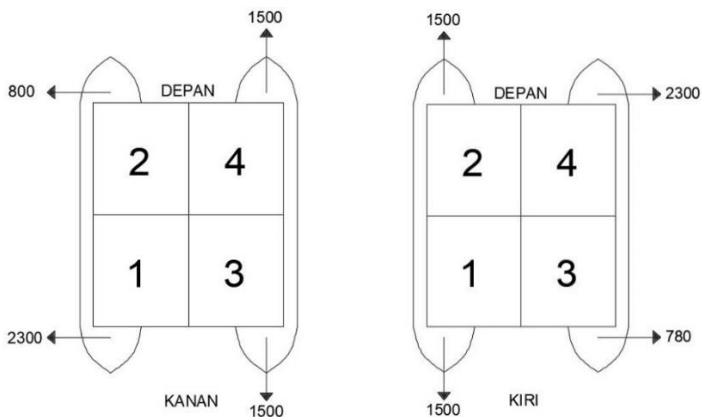


Gambar 4. 31 Respon posisi arah hadap motor kondisi Surge

Pada kondisi respon pergerakan *Surge* sudut posisi motor akan mengarah seperti pada gambar diatas dimana untuk kondisi maju motor belakang akan menghadap lurus kebelakang dengan posisi servo pada data 1500 dan motor akan berputar pada kecepatan 1700 sampai 1850 pada data PWM menyesuaikan nilai eror terhadap sumbu Y. Sedangkan untuk posisi motor dibagian depan akan menghadap ke arah samping sambil berputar pada kondisi Stand-by yaitu dengan kecepatan motor 1500 sampai 1700 pada data PWM untuk mempertahankan heading platform. Begitupula dengan kondisi mundur posisi respon pergerakan motor sama dengan kondisi maju namun pada arah yang berlawanan.

#### 4.6.3. Sway

Respon pergerakan *Sway* terjadi ketika platform menerima gaya dari arus air laut dari arah samping kanan atau kiri platform. Pergerakan sway platform terdiri dari pergerakan ke samping kanan dan kiri dari arah hadap platform. Platform akan bergerak ke samping kanan atau kiri menuju set point ketika platform menerima nilai eror delta x pada jarak rentang 1m sampai 2 m dari set point.

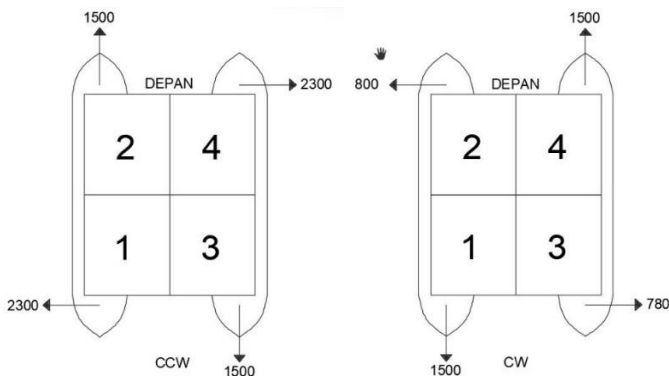


Gambar 4. 32 Respon posisi arah hadap motor kondisi *Sway*

Pada kondisi pergerakan *Sway* posisi motor thruster akan mengarah seperti pada gambar diatas dimana untuk kondisi pergerakan platform ke arah kanan posisi motor *thruster* 1 dan 2 akan mengarah pada kondisi Samping dengan nilai data servo pada gambar diatas dengan kecepatan motor pada *membership function Moving* yaitu bergerak dengan kecepatan 1700 sampai 1850 pada data PWM sesuai dengan nilai eror yang di dapat. Sedangkan untuk motor 3 dan 4 akan mengarah pada kondisi Lurus dengan data servo 1500 dan kecepatan motor 3 dan 4 pada kondisi *Stand-by* dimana rentang kecepatannya adalah 1500 sampai 1700 pada data PWM untuk mempertahankan arah *heading* Platform.

4.6.4. *Yaw*

Respon pergerakan *Yaw* terjadi ketika arah heading kapal bergeser disebabkan oleh gaya yang disebabkan arus air yang membuat platform harus bergerak secara beputar baik berputar secara *Clock Wise (CW)* dan *Conter Clock Wise (CCW)* menuju ke arah heading kapal yang sudah ditentukan. Platform akan bergerak ke arah *Clock Wise* apabila nilai eror sudut berkisar antara -15 derajat hingga -180 derajat terhadap arah heading kapal. Dan platform akan bergerak ke arah *CCW* apabila nilai eror sudut berkisar dari 15 derajat hingga 180 derajat terhadap arah heading kapal.

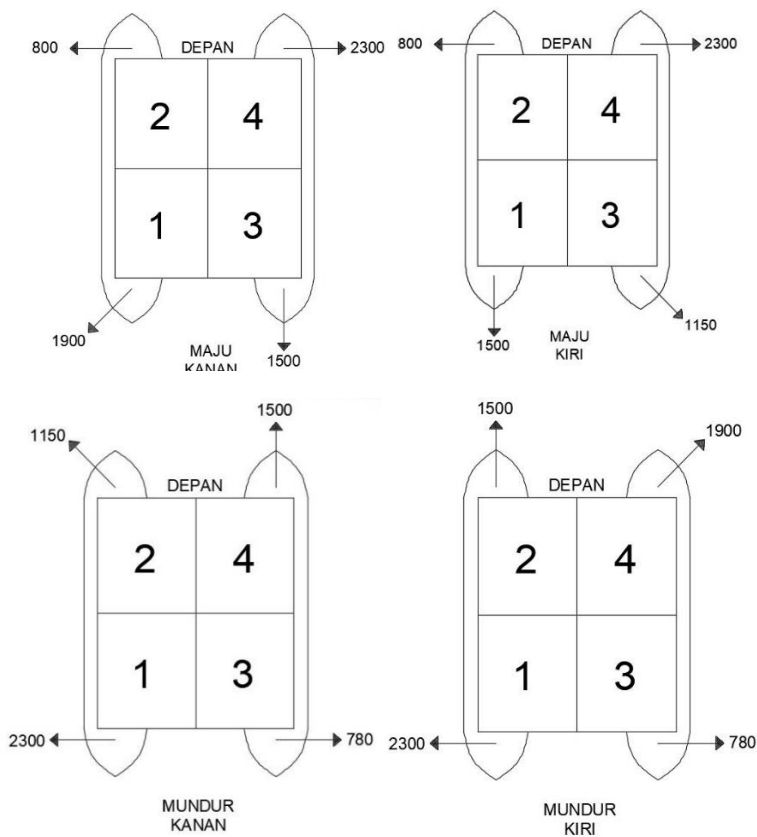


Gambar 4. 33 Respon posisi arah motor kondisi *Yaw*

Pada kondisi respon pergerakan *Yaw* posisi motor *thruster* akan mengarah seperti pada kondisi diatas, dimana untuk kondisi pergerakan CW motor thruster 2 dan 3 akan mengarah pada kondisi Samping dimana dengan data motor servo adalah 800 untuk motor 2 dan 780 untuk motor 3 pada data PWM dan kecepatan motor thruster 2 dan 3 pada kondisi Moving untuk menuju ke arah heading yang ditentukan. Sedangkan untuk posisi motor 4 dan 1 mengarah pada kondisi lurus dengan data pergerakan motor servo adalah 1500 pada data PWM dengan kecepatan motor pada kondisi *Stand-by*.

#### 4.6.5. Kombinasi *Surge* dan *Sway*

Respon pergerakan Serong atau kombinasi antara pergerakan *Surge* dan **Sway** terjadi ketika Model menerima gaya dari arus air pada posisi sisi serong platform. Platform akan bergerak ke arah serong baik itu serong maju kanan, maju kiri, mundur kanan, dan mundur ke kiri apabila platform mendapatkan nilai eror delta x dan delta y lebih dari 1m dan tidak lebih dari 2m terhadap set point yang sudah ditentukan. Pada kondisi pergerakan serong ini nilai eror sudut heading adalah pada kondisi *standby* dimana eror sudut *heading* hanya berkisar dari sudut -15 derajat sampai 15 derajat.



Gambar 4. 34 Respon posisi arah motor kondisi quarter

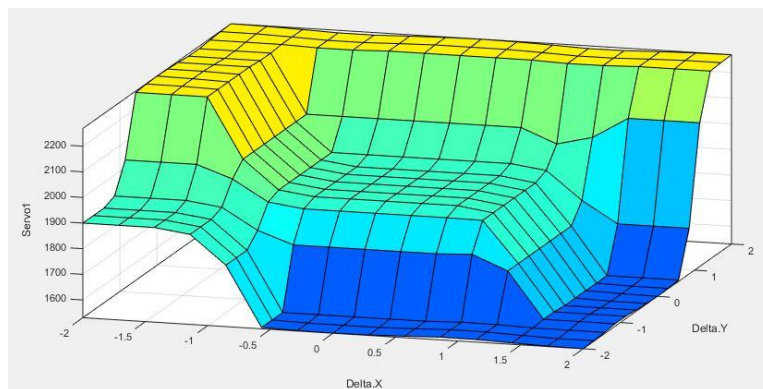
Pada kondisi pergerakan serong posisi motor *thruster* akan mengarah seperti pada gambar diatas menyesuaikan arah serong yang dituju. Ketika Model bergerak secara serong ke arah maju kanan posisi motor akan mengarah dengan data servo seperti kondisi pergerakan maju, hanya saja motor 1 berubah posisi ke kondisi Serong dengan data 1900 pada PWM. Untuk kecepatan motor, motor *thruster* 1,2, dan 3 akan bergerak pada kondisi kecepatan Moving sedangkan motor *thruster* 4 akan bergerak dengan kecepatan Stand-by dimana motor 4 difungsikan untuk memberikan *feedback* agar dapat mempertahankan posisi *heading* platform.

#### 4.7. Kurva Hubungan Input dan Output Sistem Kontrol

Berikut merupakan gambaran kurva hubungan antara input dan output yang dihasilkan dari rancangan sistem algoritma *Dynamic Positioning System* pada Model *Platform Semi-submersible*.

##### 4.7.1. Kurva Input Delta X dan Delta Y terhadap Output Servo

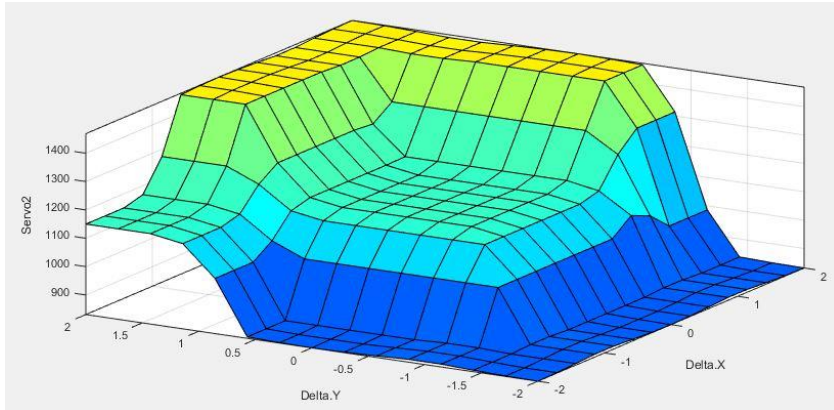
###### a. Kurva Delta X dan Delta Y terhadap Servo 1



Gambar 4. 35 Kurva Delta X dan Delta Y terhadap Servo 1

Pada gambar diatas menunjukkan kurva hubungan antara delta x dan delta Y terhadap pergerakan servo 1. Kondisi servo lurus ditunjukkan pada warna biru tua dengan data 1500 PWM, kondisi servo menghadap serong ditunjukkan pada warna hijau kebiruan dengan data 1900 PWM dan warna kuning menunjukkan kondisi servo menghadap samping dengan data 2300 PWM. Servo 1 akan berada pada kondisi serong ketika delta x dan delta y berada pada nilai -1 sampai 1m dan ketika delta x dan delta y pada nilai -2 sampai -1 m. Kondisi servo lurus terjadi ketika nilai delta x pada -1 sampai 2 m dan delta y pada -2 sampai -1 m serta delta x pada nilai 1 sampai 2 meter dan delta y pada -2 sampai 1 m. Serta kondisi servo menghadap samping terjadi ketika delta x pada nilai -2 sampai -1 m dan delta y pada -1 sampai 2m serta ketika delta x pada -2 sampai 2 m dan delta y pada 1 sampai 2 m.

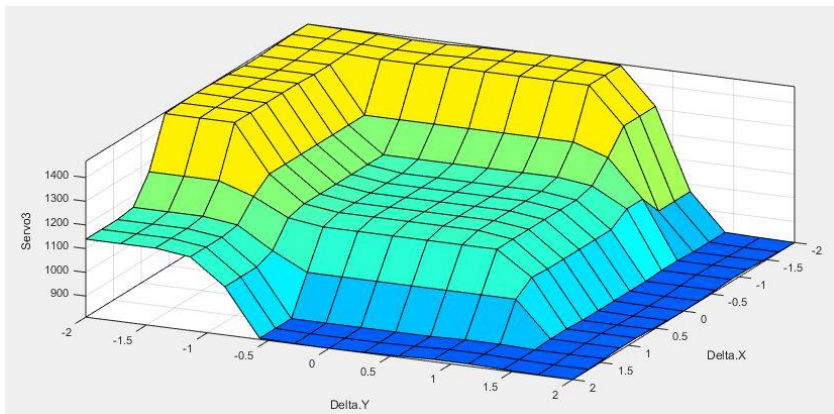
b. Kurva Delta X dan Delta Y terhadap Servo 2



Gambar 4. 36 Kurva Delta X dan Delta Y terhadap Servo 2

Pada gambar diatas menunjukkan kurva hubungan antara delta x dan delta Y terhadap pergerakan servo 2. Kondisi servo lurus ditunjukkan pada warna kuning dengan data 1500 PWM, kondisi servo menghadap serong ditunjukkan pada warna hijau kebiruan dengan data 1150 PWM dan warna kuning menunjukkan kondisi servo menghadap samping dengan data 800 PWM. Servo 2 akan berada pada kondisi serong ketika delta x dan delta y berada pada nilai -1 sampai 1m dan ketika delta x pada nilai -2 sampai -1 m dan delta y pada nilai 2 sampai 1 m. Kondisi servo lurus terjadi ketika nilai delta x pada -1 sampai 2 m dan delta y pada 2 sampai 1 m serta delta x pada nilai 1 sampai 2 meter dan delta y pada 2 sampai -1 m. Serta kondisi servo menghadap samping terjadi ketika delta x pada nilai -2 sampai -1 m dan delta y pada -1 sampai 2m serta ketika delta x pada -2 sampai 2 m dan delta y pada 1 sampai 2 m.

c. Kurva Delta X dan Dleta Y terhadap Servo 3

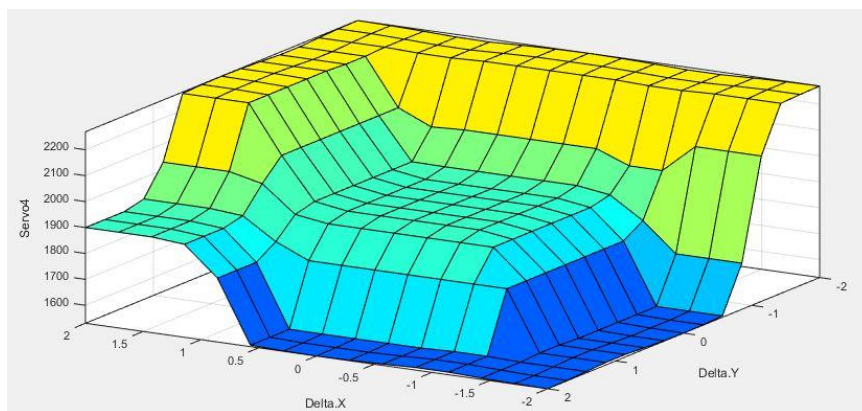


Gambar 4. 37 Kurva Delta X dan Delta Y terhadap Servo 3



Pada gambar diatas menunjukkan kurva hubungan antara delta x dan delta Y terhadap pergerakan servo 3. Kondisi servo lurus ditunjukkan pada warna kuning dengan data 1500 PWM, kondisi servo menghadap serong ditunjukkan pada warna hijau kebiruan dengan data 1140 PWM dan warna kuning menunjukkan kondisi servo menghadap samping dengan data 780 PWM. Servo 2 akan berada pada kondisi serong ketika delta x dan delta y berada pada nilai -1 sampai 1m dan ketika delta x pada nilai 2 sampai 1 m dan delta y pada nilai -2 sampai -1 m. Kondisi servo lurus terjadi ketika nilai delta x pada 1 sampai -2 m dan delta y pada -2 sampai -1 m serta delta x pada nilai -1 sampai -2 meter dan delta y pada -2 sampai 1 m. Serta kondisi servo menghadap samping terjadi ketika delta x pada nilai 2 sampai 1 m dan delta y pada -1 sampai 2 m serta ketika delta x pada -2 sampai 2 m dan delta y pada 1 sampai 2 m.

d. Kurva Delta X dan Delta Y terhadap Servo 4

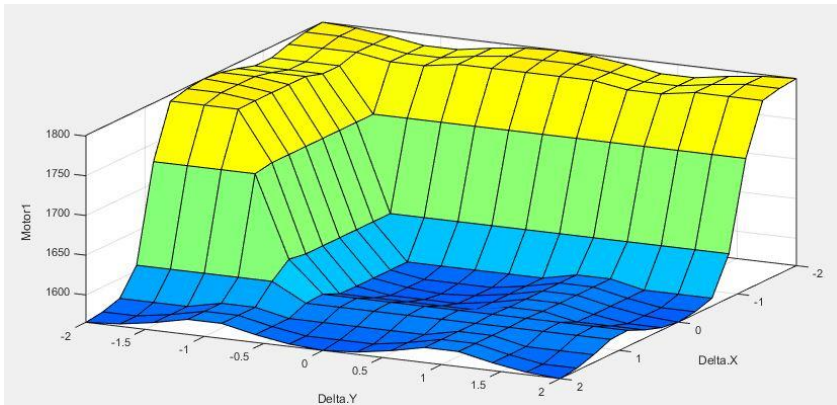


Gambar 4. 38 Kurva Delta X dan Delta Y terhadap Servo 4

Pada gambar diatas menunjukkan kurva hubungan antara delta x dan delta Y terhadap pergerakan servo 4. Kondisi servo lurus ditunjukkan pada warna biru tua dengan data 1500 PWM, kondisi servo menghadap serong ditunjukkan pada warna hijau kebiruan dengan data 1900 PWM dan warna kuning menunjukkan kondisi servo menghadap samping dengan data 2300 PWM. Servo 4 akan berada pada kondisi serong ketika delta x dan delta y berada pada nilai -1 sampai 1m dan ketika delta x dan delta y pada nilai 2 sampai 1 m. Kondisi servo lurus terjadi ketika nilai delta x pada 1 sampai -2 m dan delta y pada 2 sampai 1 m serta delta x pada nilai -1 sampai -2 meter dan delta y pada 2 sampai -1 m. Serta kondisi servo menghadap samping terjadi ketika delta x pada nilai 2 sampai 1 m dan delta y pada 1 sampai -2 m serta ketika delta x pada -2 sampai 2 m dan delta y pada -1 sampai -2 m.

#### 4.7.2. Kurva Input Delta X dan Delta Y terhadap Output Motor.

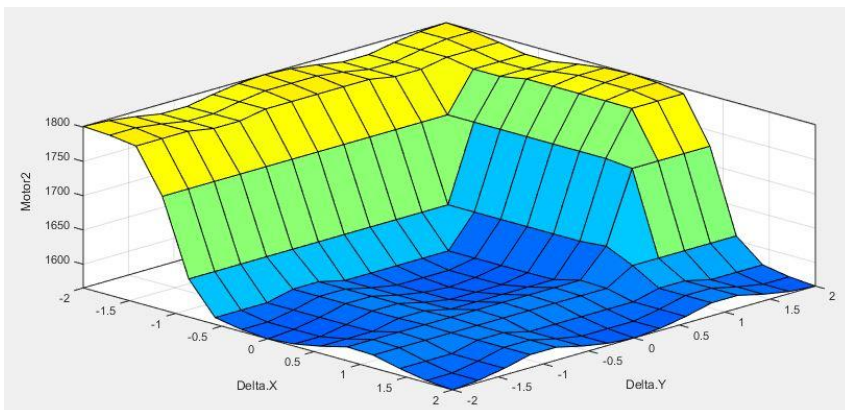
##### a. Kurva Delta X dan Delta Y terhadap Motor 1



Gambar 4. 39 Kurva Delta X delta Y terhadap Motor 1

Pada gambar kurva diatas menunjukkan hubungan antar delta x dan delta y terhadap output motor 1. Warna biru pada kurva diatas menunjukkan kondisi motor pada kecepatan *standby* dengan rentang data kecepatan motor adalah 1500 sampai 1700 PWM. Sedangkan biru pada krva menunjukkan kondisi motor pada kecepatan *Moving* dengan rentang data kecepatan motor adalah 1700 sampai 1850 PWM. Motor 1 akan bergerak pada kecepatan *Stand-by* ketika nilai delta x pada -1 sampai 2 m dan delta y pada -1 sampai 2 m serta ketika delta x pada nilai 1 sampai 2 m dan delta y pada -2 sampai -1 m. Motor 1 akan bergerak pada kecepatan *Moving* ketika nilai delta x pada 1 sampai -2 m dan delta y pada -2 sampai -1 m serta ketika delta x pada -1 sampai -2 m dan delta y pada -2 sampai 2 m.

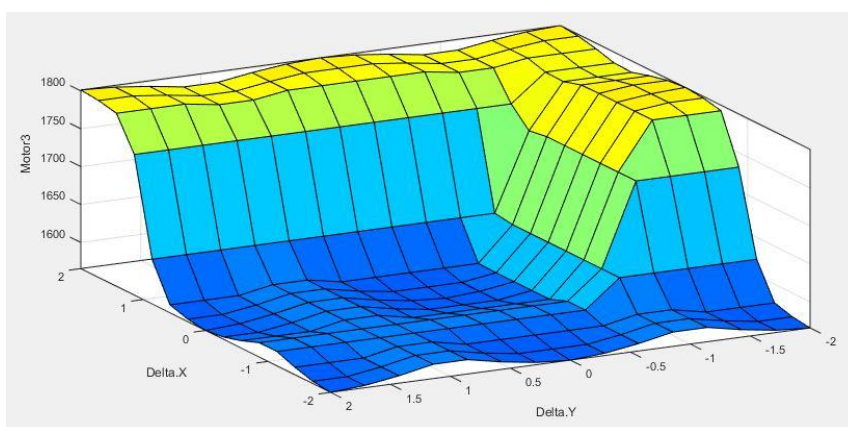
##### b. Kurva Delta X dan Delta Y terhadap Motor 2



Gambar 4. 40 Kurva Delta X dan Delta Y terhadap Motor 2

Pada gambar kurva diatas menunjukkan hubungan antar delta x dan delta y terhadap output motor 2. Warna biru pada kurva diatas menunjukkan kondisi motor pada kecepatan *standby* dengan rentang data kecepatan motor adalah 1500 sampai 1700 PWM. Sedangkan biru pada krva menunjukkan kondisi motor pada kecepatan *Moving* dengan rentang data kecepatan motor adalah 1700 sampai 1850 PWM. Motor 2 akan bergerak pada kecepatan *Stand-by* ketika nilai delta x pada -1 sampai 2 m dan delta y pada -2 sampai 1 m serta ketika delta x pada nilai 1 sampai 2 m dan delta y pada 1 sampai 2 m. Motor 2 akan bergerak pada kecepatan *Moving* ketika nilai delta x pada 1 sampai -2 m dan delta y pada 2 sampai 1 m serta ketika delta x pada -1 sampai -2 m dan delta y pada -2 sampai 2 m.

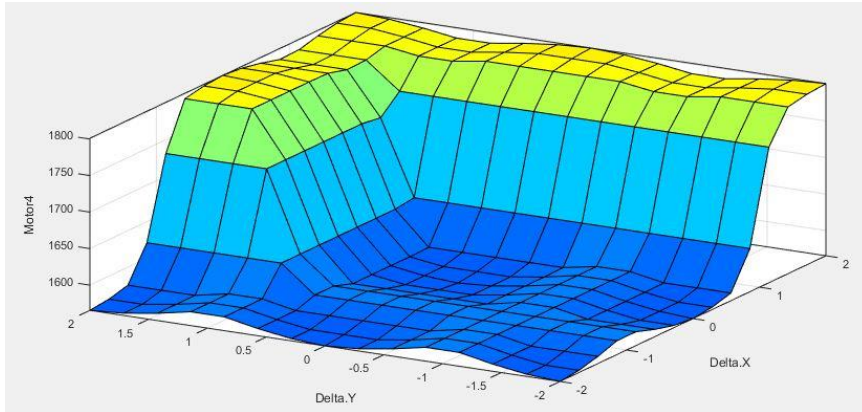
c. Kurva Delta X dan Delta Y terhadap Motor 3



Gambar 4. 41 Kurva Delta X dan Delta Y terhadap Motor 3

Pada gambar kurva diatas menunjukkan hubungan antar delta x dan delta y terhadap output motor 3. Warna biru pada kurva diatas menunjukkan kondisi motor pada kecepatan *standby* dengan rentang data kecepatan motor adalah 1500 sampai 1700 PWM. Sedangkan biru pada krva menunjukkan kondisi motor pada kecepatan *Moving* dengan rentang data kecepatan motor adalah 1700 sampai 1850 PWM. Motor 3 akan bergerak pada kecepatan *Stand-by* ketika nilai delta x pada -2 sampai 1 m dan delta y pada -1 sampai 2 m serta ketika delta x pada nilai -1 sampai -2 m dan delta y pada -1 sampai -2 m. Motor 3 akan bergerak pada kecepatan *Moving* ketika nilai delta x pada -1 sampai 2 m dan delta y pada -2 sampai -1 m serta ketika delta x pada 1 sampai 2 m dan delta y pada -2 sampai 2 m.

d. Kurva Delta X dan Delta Y terhadap Motor 4

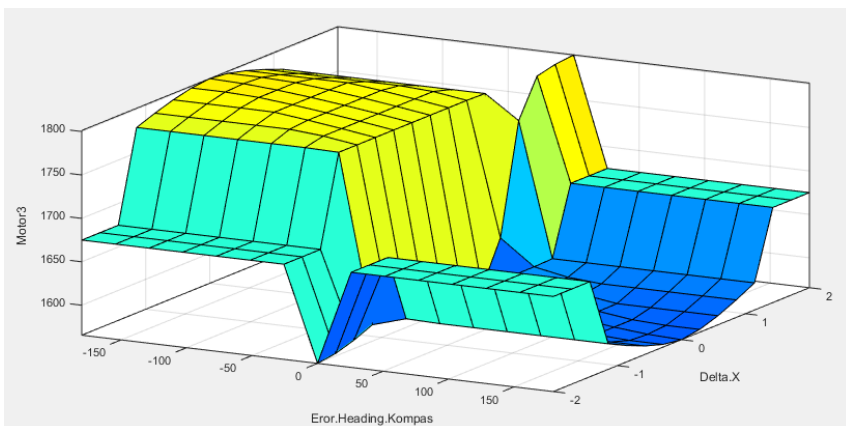


Gambar 4. 42 Kurva Delta X dan Delta Y terhadap Motor 4

Pada gambar kurva diatas menunjukkan hubungan antar delta x dan delta y terhadap output motor 4. Warna biru pada kurva diatas menunjukkan kondisi motor pada kecepatan *standby* dengan rentang data kecepatan motor adalah 1500 sampai 1700 PWM. Sedangkan biru pada krva menunjukkan kondisi motor pada kecepatan *Moving* dengan rentang data kecepatan motor adalah 1700 sampai 1850 PWM. Motor 4 akan bergerak pada kecepatan *Stand-by* ketika nilai delta x pada -2 sampai 1 m dan delta y pada -2 sampai 1 m serta ketika delta x pada nilai -1 sampai -2 m dan delta y pada 1 sampai 2 m. Motor 4 akan bergerak pada kecepatan *Moving* ketika nilai delta x pada -1 sampai 2 m dan delta y pada 2 sampai 1 m serta ketika delta x pada 1 sampai 2 m dan delta y pada -2 sampai 2 m.

#### 4.7.3. Kurva Input Delta X dan Error Heading terhadap Output Motor dan Servo

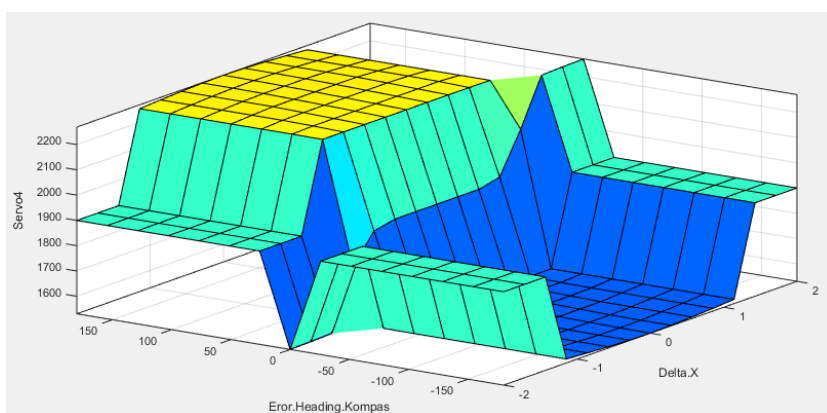
a. Kurva Delta X dan Error Heading terhadap Motor



Gambar 4. 43 Kurva Delta X dan Error Heading terhadap Motor

Pada gambar kurva diatas menunjukkan salah satu contoh hubungan antara nilai input delta X dan eror *Heading* terhadap output berupa kecepatan motor 3. Pada gambar diatas warna biru tua menunjukkan kondisi motor bergerak pada kecepatan *Stand-by* dengan rentang kecepatan 1500-1700 PWM. Sedangkan warna kuning menunjukkan motor bergerak dengan kondisi kecepatan *moving* yaitu pada rentang kecepatan 1700 sampai 1850 PWM. Pada gambar kurva diatas, motor 3 akan bergerak pada kecepatan *stand-by* ketika eror *heading* lebih dari 15 derajat dan ketika delta x pada kondisi *stand-by* yaitu pada -1 sampai 1m. Dan motor 4 akan bergerak pada kondisi *Moving* ketika nilai eror heading kurang dari -15 derajat dan nilai delta x pada kondisi *stand-by* juga yaitu pada -1 sampai 1 m.

b. Kurva delta X dan eror Heading terhadap Servo

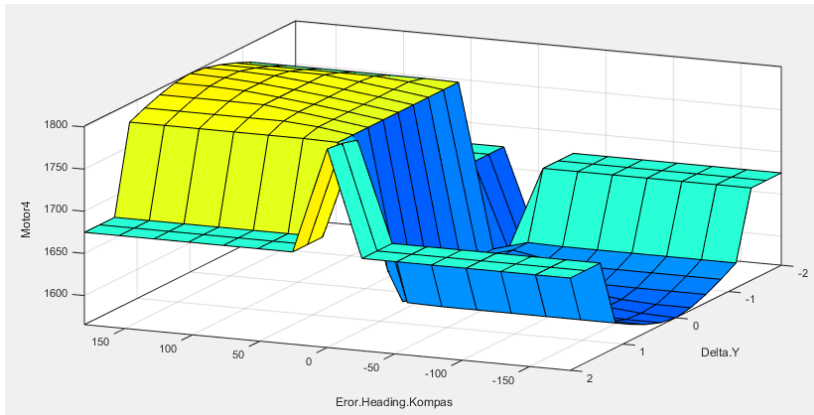


Gambar 4. 44 Kurva Delta X dan eror Heading terhadap Servo 4

Gambar Kurva diatas merupakan salah satu contoh hubungan antara eror *heading* dan delta X terhadap output Servo. Warna kuning datar pada gambar kurva diatas menunjukkan kondisi servo 4 menghadap Samping dengan data 2300 PWM. Warna biru menunjukkan kondisi servo 4 menghadap Lurus dengan data 1500 PWM. Servo 4 akan menghadap e samping ketika nilai eror *heading* lebih dari 15 derajat dan delta x berada pada nilai *stand-by* yaitu 1 sampai -1 m. Dan Servo 4 akan berada pada kondisi lurus ketika eror heading kompas kurang dari -15 derajat dan delta x berada pada kondisi *Stand-by* yaitu -1 sampai 1m.

#### 4.7.4. Kurva Input Delta Y dan Error Heading Terhadap motor dan Servo.

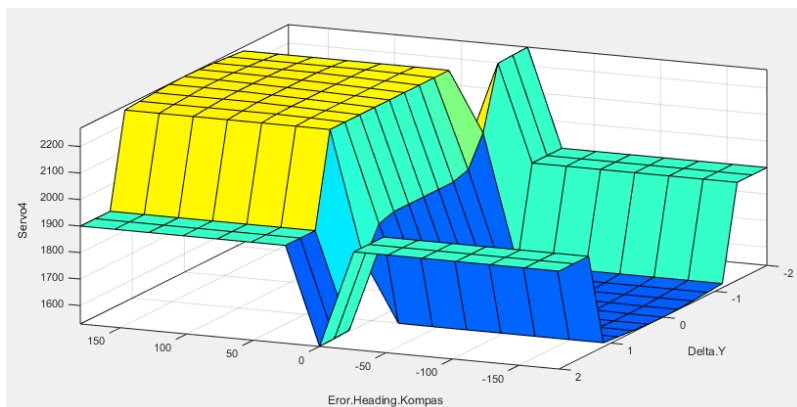
##### a. Kurva Delta Y dan Error Heading Terhadap Motor.



Gambar 4. 45 Kurva Delta Y dan Error Heading terhadap Motor 4.

Pada gambar kurva diatas menunjukkan salah satu contoh hubungan antara nilai inputan delta Y dan error *Heading* terhadap output berupa kecepatan motor 4. Pada gambar diatas warna biru tua menunjukkan kondisi motor bergerak pada kecepatan *Stand-by* dengan rentang kecepatan 1500-1700 PWM. Sedangkan warna kuning menunjukkan motor bergerak dengan kondisi kecepatan *moving* yaitu pada rentang kecepatan 1700 sampai 1850 PWM. Pada gambar kurva diatas, motor 4 akan bergerak pada kecepatan *stand by* ketika error heading lebih dari 15 derajat dan ketika delta x pada kondisi *stand by* yaitu pada -1 sampai 1m. Dan motor 4 akan bergerak pada kondisi *Moving* ketika nilai error *heading* kurang dari -15 derajat dan nilai delta x pada kondisi *stand-by* juga yaitu pada -1 sampai 1 m.

##### b. Kurva Delta Y dan Error Heading Terhadap Servo



Gambar 4. 46 Kurva Delta Y dan Error Heading terhadap Servo 4.

Gambar Kurva diatas merupakan salah satu contoh hubungan antara eror *heading* dan delta Y terhadap output Servo. Warna kuning datar pada gambar kurva diatas menunjukkan kondisi servo 4 menghadap Samping dengan data 2300 PWM. Warna biru menunjukkan kondisi servo 4 menghadap Lurus dengan data 1500 PWM. Servo 4 akan menghadap ke samping ketika nilai eror *heading* lebih dari 15 derajat dan delta x berada pada nilai stand by yaitu 1 sampai -1 m. Dan Servo 4 akan berada pada kondisi lurus ketika eror *heading* kompas kurang dari -15 derajat dan delta x berada pada kondisi *Stand-by* yaitu -1 sampai 1m.

#### 4.8. Analisa Hasil Uji Coba

##### 4.8.1. Uji Coba Manual Model

Uji coba yang pertama dilakukan adalah proses pengujian manual Model Semi-Submersible. Uji coba manual dilakukan untuk mengetahui apakah sistem mekanik dan propulsi pada platform berjalan dengan baik. Kemudian untuk mengetahui sistem elektronik yang terpasang di platform sudah bisa difungsikan dengan baik. Dan yang terakhir tujuan dilakukan uji coba manual adalah untuk mengatui skenario pergerakan Model *Platform Semi-submersible* dengan menggunakan 4 *azimuth thruster* sudah berjalan dengan baik.

Proses uji coba manual dilakukan di danau delapan ITS. Kontrol pengujian manual pada platform dilakukan dengan menggunakan bantuan remot kontrol. Sehingga platform bisa dijangkau pada jarak yang cukup jauh. Berikut merupakan dokumentasi proses pengujian yang dilakukan di danau delapan ITS.



Gambar 4. 47 Uji Coba sistem Propulsi dan Elektronika Model



Gambar 4. 48 Pengujian Pergerakan Model

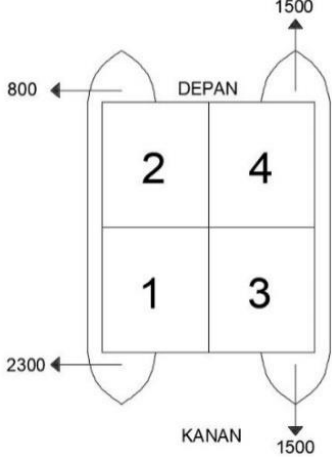
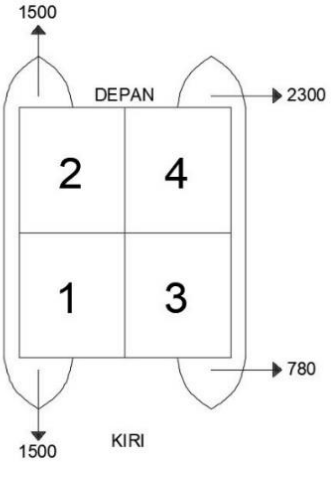
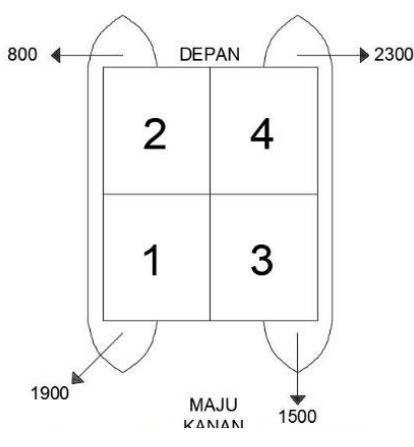
Dari hasil pengujian pergerakan Model yang terdiri dari 10 jenis pergerakan yang mengacu pada 3 DOF (*Degree of freedom*) sudah berjalan dengan baik, namun masih terdapat sedikit kendala pada pergerakan mundur dan serong, hal ini terjadi dikarenakan pada posisi motor yang kurang tepat akibat sistem mekanisme propulsi yang digunakan sebelumnya menggunakan belt untuk transmisi antara motor servo dengan motor thruster. Untuk mengatasi masalah tersebut sistem transmisi diganti dengan menggunakan gear sehingga mengurangi terjadinya slip yang membuat posisi motor thruster bisa lebih akurat.

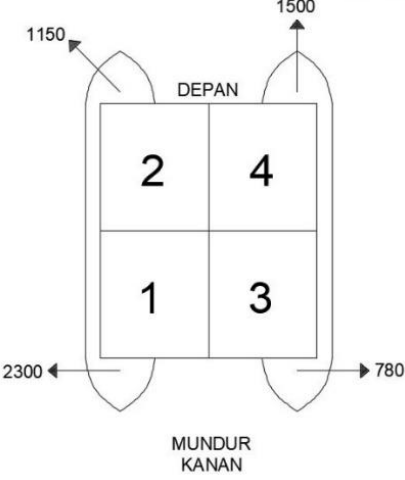
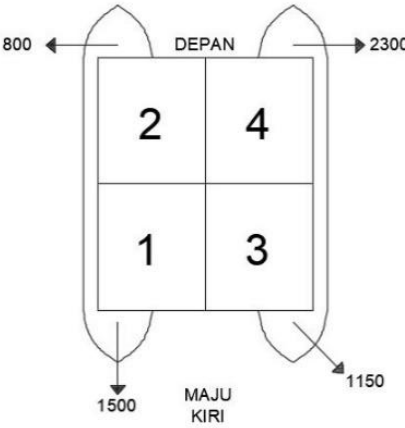
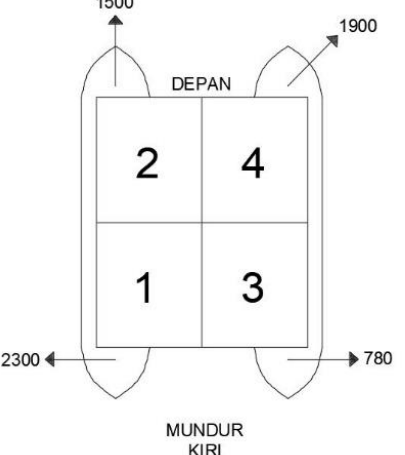
Berikut merupakan tabel analisa hasil pengujian Model secara manual dengan menggunakan kontrol jarak jauh menggunakan remot kontrol.

Tabel 4. 4 Hasil Uji Coba Pergerakan Manual

No	Pergerakan	Skema Pergerakan	Hasil	Keterangan
1	Maju		Baik	<p>Motor 1 = On,            Motor 3 = On,            Motor 2 = Off,            Motor 4 = Off,            Posisi Servo = sesuai dengan skema pergerakan,            Trim = -</p>
2	Mundur		Baik	<p>Motor 1 = Off,            Motor 3 = Off,            Motor 2 = On,            Motor 4 = On,            Posisi Servo = sesuai dengan skema pergerakan,            Trim = -</p>



No	Pergerakan	Skema Pergerakan	Hasil	Keterangan
3	Kanan		Baik	<p>Motor 1 = On,            Motor 3 = Off,            Motor 2 = On,            Motor 4 = Off,            Posisi Servo =            sesuai dengan            skema            pergerakan,            Trim = -</p>
4	Kiri		Baik	<p>Motor 1 = Off,            Motor 3 = On,            Motor 2 = Off,            Motor 4 = On,            Posisi Servo =            sesuai dengan            skema            pergerakan,            Trim = -</p>
5	Maju Kanan		Baik	<p>Motor 1 = On,            Motor 3 = On,            Motor 2 = Off,            Motor 4 = Off,            Posisi Servo =            sesuai dengan            skema            pergerakan,            Trim = -</p>

No	Pergerakan	Skema Pergerakan	Hasil	Keterangan
6	Mundur Kanan		Baik	<p>Motor 1 = Off,          Motor 3 = Off,          Motor 2 = On,          Motor 4 = On,          Posisi Servo = sesuai dengan skema pergerakan,          Trim = -</p>
7	Maju Kiri		Baik	<p>Motor 1 = On,          Motor 3 = On,          Motor 2 = Off,          Motor 4 = Off,          Posisi Servo = sesuai dengan skema pergerakan,          Trim = -</p>
8	Mundur Kiri		Baik	<p>Motor 1 = Off,          Motor 3 = Off,          Motor 2 = On,          Motor 4 = On,          Posisi Servo = sesuai dengan skema pergerakan,          Trim = -</p>

No	Pergerakan	Skema Pergerakan	Hasil	Keterangan
9	Pivot CW		Baik	<p>Motor 1 = Off,          Motor 3 = On,          Motor 2 = On,          Motor 4 = Off,          Posisi Servo =          sesuai dengan          skema          pergerakan,          Trim = -</p>
10	Pivot CCW		Baik	<p>Motor 1 = On,          Motor 3 = Off,          Motor 2 = Off,          Motor 4 = On,          Posisi Servo =          sesuai dengan          skema          pergerakan,          Trim = -</p>

#### 4.8.2. Uji Coba Automatis Model

Pengujian otomatis Model dilakukan dengan tujuan untuk mengetahui apakah Model platform semi-submersible dengan *Dynamic Positioning Sistem* dapat berjalan secara otomatis dengan baik, mengetahui respon pergerakan terhadap pergeseran platform serta respon kecepatan motor terhadap jarak pergeseran platform. Proses pengujian Model dilakukan sesuai dengan skenario yang sudah ditentukan yang terdiri dari 4 skenario. Skenario tersebut meliputi skenario uji coba pergerakan dengan pergeseran secara *Surgee*, *Sway*, *Yaw*, dan kombinasi antara *Surge* dan *Sway*.

Langkah awal pada pengujian *Dynamic Positioning System Model platform Semi-submersible* dengan *Metode DGPS (Differential Global Positioning System)* adalah dengan menentukan titik-titik set point atau titik tujuan platform, titik *Refrensi Based* di darat dan menentukan arah heading platform. Untuk arah heading platform adalah pada posisi 0 derajat sedangkan titik-titik *set point* dan *refrensi* adalah sebagai berikut.

Tabel 4. 5 Data Titik *Set Point* Dan Titik Refrensi Model

Titik	<i>Latitude</i>	<i>Longitude</i>	Jarak (m)
Set Point	-7.286853	112.79598	7.23
Based Refrence Point	-7.286878	112.79592	



Gambar 4. 49 Penentuan Set Point dan Heading Platform

Setelah menentukan titik-titik utama pada Model, kemudian Model dilakukan ujicoba secara otomatis dengan melakukan pergeseran platform pada 8 titik pergeseran dan 3 arah perputaran untuk mengetahui respon dri DPS yang sudah diterapkan pada Model platform.

#### A. Uji Coba Pergerakan Surgee

Sesuai dengan skenario pengujian yang sudah dibuat, proses ujicoba pergerakan Surgee dilakukan dengan menggeser platform dari set point dengan arah pergeseran Surge terhadap heading platform. Berikut merupakan data titik pergeseran platform.

Tabel 4. 6 Data ujicoba pergerakan Surgee Heading 0°

Titik	<i>Latitude</i>	<i>Longitude</i>	Jarak (m)
Set Point	-7.286853	112.79598	0
Pergeseran Maju	-7.286834913	112.7959809	2
Pergeseran Mundur	-7.286870966	112.7959791	2
Pergeseran	Respon Pergerakan	Eror ke Set Point (m)	Waktu (s)
Maju	Mundur	0.69	23
Mundur	Maju	0.59	39

Dari hasil ujicoba pergeseran platform untuk mengetahui respon pergerakannya didapatkan bahwa platform dapat bergerak sesuai dengan respon yang dirancang. Ketika digeser maju dari set point platform akan bergerak mundur kembali menuju ke set point begitu pula dengan pergeseran mundur. Namun pada pengujian pergerakan Surgee nilai eror jarak dari platform ke set point masih lebih dari 0.5 m. Hal ini disebabkan karena data dri sensor GPS yang selalu berubah-ubah atau mengalami drifting.

Dari pengujian tersebut juga didapatkan data respon kecepatan motor pada setiap respon pergerakan, dimana data respon kecepatan motor didapatkan langsung dari desain fuzzy logic di Matlab sebagai berikut.

a. Pergeseran Maju

Tabel 4. 7 Data respon kecepatan motor pergeseran maju

Jarak (m)	Kecepatan Motor							
	Motor 1		Motor 2		Motor 3		Motor 4	
	PWM	RPM	PWM	RPM	PWM	RPM	PWM	RPM
2	1570	1040	1850	5200	1570	1040	1850	5200
1.5	1570	1040	1800	4457	1570	1040	1800	4457
1	1590	1337	1670	2526	1590	1337	1670	2526
0.5	1570	1040	1570	1040	1570	1040	1570	1040

Tabel 4. 8 Data servo gerak mundur

Data Servo Mundur		
Servo	PWM	Sudut
1	2300	90
2	1500	0
3	780	90
4	1500	0

Pada pengujian pergeseran maju, respon pergerakan platform adalah bergerak mundur. Pada pergerakan mundur motor yang akan bergerak dalam kondisi moving adalah motor 2 dan motor 4, sedangkan motor 1 dan 3 dalam kondisi kecepatan stand-by. Pada data yang didapatkan dri simulasi matlab didapatkan respon bahwa semakin jauh jarak pergeseran platform terhadap setpoint kecepatan motor akan semakin tinggi.

b. Pergeseran Mundur

Tabel 4. 9 Data respon kecepatan motor pergeseran mundur

Jarak (m)	Kecepatan Motor							
	Motor 1		Motor 2		Motor 3		Motor 4	
	PWM	RPM	PWM	RPM	PWM	RPM	PWM	RPM
2	1850	5200	1570	1040	1850	5200	1570	1040
1.5	1800	4457	1570	1040	1800	4457	1570	1040

Jarak (m)	Kecepatan Motor							
	Motor 1		Motor 2		Motor 3		Motor 4	
	PWM	RPM	PWM	RPM	PWM	RPM	PWM	RPM
1	1670	2526	1590	1337	1670	2526	1590	1337
0.5	1570	1040	1570	1040	1570	1040	1570	1040

Tabel 4. 10 Data servo gerak maju

Data Servo Maju		
Servo	PWM	Sudut
1	1500	0
2	800	90
3	2300	90
4	1500	0

Pada pengujian pergeseran mundur, respon pergerakan platform adalah bergerak maju. Pada pergerakan maju motor yang akan bergerak pada kecepatan moving adalah motor 1 dan 3 sedangkan motor 2 dan 4 pada kondisi kecepatan Stand-by. Pada data yang di dapatkan dari simulasi matlab didapatkan respon bahwa semakin jauh jarak pergeseran maka kecepatan motor akan semakin tinggi.

#### B. Uji Coba Pergerakan Sway

Sesuai dengan skenario ujicoba yang sudah dirancang, ujicoba pergerakan Sway dilakukan dengan menggeser platform searah pergerakan Sway dari set point terhadap hading platform. Berikut merupakan data titik pergeseran platform secara sway.

Tabel 4. 11 Data ujicoba pergerakan Sway Heading 0°

Titik	<i>Latitude</i>	<i>Longitude</i>	Jarak (m)
Set Point	-7.286853	112.79598	0
Pergeseran Kanan	-7.286852096	112.795998	2
Pergeseran Kiri	-7.286853904	112.795962	2
Pergeseran	Respon Pergerakan	Error ke Set Point (m)	Waktu (s)
Kanan	Kiri	0.31	25
Kiri	Kanan	0.88	28

Dari hasil ujicoba pergeseran platform untuk mengetahui respon pergerakannya didapatkan bahwa platform dapat bergerak sesuai dengan respon yang dirancang. Ketika digeser ke kanan dari set point platform akan bergerak ke kiri kembali menuju ke set point begitu pula dengan pergeseran ke kiri. Namun pada pengujian pergerakan Surgee nilai eror jarak dari platform ke set point masih lebih dari 0.5 m, yaitu pada pergerakan ke kanan. Hal ini disebabkan karena data dri sensor GPS yang selalu berubah-ubah atau mengalami drifting.

Dari pengujian tersebut juga didapatkan data respon kecepatan motor pada setiap respon pergerakan, dimana data respon kecepatan motor didapatkan langsung dari desain fuzzy logic di Matlab sebagai berikut.

a. Pergeseran Kanan

Tabel 4. 12 Data respon kecepatan motor pergeseran kanan

Jarak (m)	Kecepatan Motor							
	Motor 1		Motor 2		Motor 3		Motor 4	
	PWM	RPM	PWM	RPM	PWM	RPM	PWM	RPM
2	1570	1040	1570	1040	1850	5200	1850	5200
1.5	1570	1040	1570	1040	1800	4457	1800	4457
1	1590	1337	1590	1337	1670	2526	1670	2526
0.5	1570	1040	1570	1040	1570	1040	1570	1040

Tabel 4. 13 Data servo gerak kiri

Data Servo Kiri		
Servo	PWM	Sudut
1	1500	0
2	1500	0
3	2300	90
4	780	90

Pada pengujian pergeseran kanan, respon pergerakan platform adalah bergerak ke kiri. Pada pergerakan ke kiri motor yang akan bergerak dalam kondisi moving adalah motor 3 dan motor 4, sedangkan motor 1 dan 2 dalam kondisi kecepatan stand-by. Pada data yang didapatkan dari simulasi matlab didapatkan respon bahwa semakin jauh jarak pergeseran platform terhadap setpoint kecepatan motor akan semakin tinggi.

b. Pergeseran Kiri

Tabel 4. 14 Data respon kecepatan motor pergeseran maju

Jarak (m)	Kecepatan Motor							
	Motor 1		Motor 2		Motor 3		Motor 4	
	PWM	RPM	PWM	RPM	PWM	RPM	PWM	RPM
2	1850	5200	1850	5200	1570	1040	1570	1040
1.5	1800	4457	1800	4457	1570	1040	1570	1040
1	1670	2526	1670	2526	1590	1337	1590	1337
0.5	1570	1040	1570	1040	1570	1040	1570	1040

Tabel 4. 15 Data servo gerak kanan.

Data Servo Kanan		
Servo	PWM	Sudut
1	2300	90
2	800	90
3	1500	0
4	1500	0

Pada pengujian pergeseran Kiri, respon pergerakan platform adalah bergerak ke Kanan. Pada pergerakan kanan motor yang akan bergerak dalam kondisi moving adalah motor 1 dan motor 2, sedangkan motor 3 dan 4 dalam kondisi kecepatan stand-by. Pada data yang didapatkan dari simulasi matlab didapatkan respon bahwa semakin jauh jarak pergeseran platform terhadap setpoint kecepatan motor akan semakin tinggi.

### C. Uji Coba Pergerakan Yaw

Sesuai dengan skenario ujicoba yang sudah dirancang, ujicoba pergerakan Yaw dilakukan dengan menggeser platform searah pergerakan yaw dari sudut heading platform pada set point yang sudah ditentukan. Berikut merupakan data pergeseran sudut platform pada set point terhadap sudut heading awalnya.

Tabel 4. 16 Data ujicoba pergerakan Yaw Heading 0°

Titik	Sudut		
Set Point Sudut	0°		
Pergeseran sudut CW	90°		
Pergeseran sudut CW	180°		
Pergeseran sudut CW	270°		
Pergeseran	Respon Pergerakan	Eror Sudut <i>Heading</i>	Waktu (s)
90°	CCW	7°	27
180°	CW	11°	29
270°	CW	7°	29

Dari hasil ujicoba pergeseran platform untuk mengetahui respon pergerakannya didapatkan bahwa platform dapat bergerak sesuai dengan respon yang dirancang. Ketika diputar ke arah Clock Wise 90 derajat dari sudut heading, platform akan bergerak memutar ke arah CCW kembali menuju ke sudut heading awal yang sudah ditentukan. begitu pula dengan pergeseran sudut heading pada sudut 180 derajat dan sudut 270 derajat. Pada pengujian pergerakan Yaw nilai eror sudut heading platform saat kembali sudah berada dibawah 15 derajat.

Dari pengujian tersebut juga didapatkan data respon kecepatan motor pada setiap respon pergerakan, dimana data respon kecepatan motor didapatkan langsung dari desain fuzzy logic di Matlab sebagai berikut.



Tabel 4. 17 Data respon kecepatan motor pergeseran Yaw

Sudut	Kecepatan Motor (PWM)							
	Motor 1		Motor 2		Motor 3		Motor 4	
	PWM	RPM	PWM	RPM	PWM	RPM	PWM	RPM
90	1790	4309	1580	1189	1580	1189	1790	4309
180	1570	1040	1800	4457	1800	4457	1570	1040
270	1580	1189	1790	4309	1790	4309	1580	1189

Tabel 4. 18 Data servo gerak CCW

Data Servo CCW		
Servo	PWM	Sudut
1	2300	90
2	1500	0
3	1500	0
4	2300	90

Tabel 4. 19 Data servo gerak CW

Data Servo CW		
Servo	PWM	Sudut
1	1500	0
2	800	90
3	780	90
4	1500	0

Pada uji coba pergeseran platform secara Yaw platform dapat bergerak sesuai dengan respon yang telah dirancang. Dari data respon kecepatan motor pada table diatas dapat tampak bahwa semakin jauh pergeseran sudut dari arah heading platform kecepatan motor akan semakin bertambah.

#### D. Uji Coba Pergerakan Kombinasi Surgee dan Sway

Sesuai dengan skenario ujicoba yang sudah dirancang, ujicoba pergerakan kombinasi Surge dan Sway dilakukan dengan menggeser platform searah pergerakan serong (Sway dan Surgee) dari set point terhadap hading platform. Berikut merupakan data titik pergeseran platform secara serong.

Tabel 4. 20 Data ujicoba pergerakan kombinasi Surge dan Sway Heading 0°

Titik	<i>Latitude</i>	<i>Longitude</i>	Jarak (m)
Set Point	-7.286853	112.79598	0
Pergeseran Maju Kanan	-7.286834913	112.795998	2
Pergeseran Maju Kiri	-7.286834913	112.795962	2
Pergeseran Mundur Kanan	-7.286871087	112.795998	2

Titik	Latitude	Longitude	Jarak (m)
Pergeseran Mundur Kiri	-7.286871087	112.795962	2
Pergeseran	Respon Pergerakan	Error ke Set Point (m)	Waktu (s)
Maju Kanan	Mundur Kiri	0.92	70
Maju Kiri	Mundur Kanan	0.23	38
Mundur Kanan	Maju Kiri	0.22	45
Mundur Kiri	Maju Kanan	0.32	48

Dari hasil ujicoba pergeseran platform untuk mengetahui respon pergerakannya didapatkan bahwa platform dapat bergerak sesuai dengan respon yang dirancang. Ketika digeser kearah maju kanan dari set point, platform akan bergerak kearah mundur kiri kembali menuju ke set point begitu pula dengan pergeseran serong yang lainnya. Namun pada pengujian pergerakan kombinasi Surgee dan sway nilai error jarak dari platform ke set point masih lebih dari 0.5 m, yaitu pada pergerakan kearah mundur kiri. Hal ini disebabkan karena data dri sensor GPS yang selalu berubah-ubah atau mengalami drifting.

Dari pengujian tersebut juga didapatkan data respon kecepatan motor pada setiap respon pergerakan, dimana data respon kecepatan motor didapatkan langsung dari desain fuzzy logic di Mathlab sebagai berikut.

a. Pergeseran Maju Kanan

Tabel 4. 21 Data respon kecepatan motor pergeseran maju kanan

Jarak (m)	Kecepatan Motor (PWM)							
	Motor 1		Motor 2		Motor 3		Motor 4	
	PWM	RPM	PWM	RPM	PWM	RPM	PWM	RPM
2	1570	1040	1570	1040	1850	5200	1850	5200
1.5	1570	1040	1570	1040	1800	4457	1800	4457
1	1590	1337	1590	1337	1670	2526	1670	2526
0.5	1570	1040	1570	1040	1570	1040	1570	1040

Tabel 4. 22 Data servo gerak mundur kiri

Data Servo Mundur Kiri		
Servo	PWM	Sudut
1	2300	90
2	1500	0
3	780	90
4	1900	45

Pada pengujian pergeseran maju kanan, respon pergerakan platform adalah bergerak mundur kiri. Pada pergerakan mundur kiri motor yang akan bergerak dalam kondisi moving adalah motor 3 dan motor 4, sedangkan motor 1 dan 2 dalam kondisi

kecepatan stand-by. Pada data yang didapatkan dari simulasi matlab didapatkan respon bahwa semakin jauh jarak pergeseran platform terhadap setpoint kecepatan motor akan semakin tinggi.

b. Pergeseran Maju Kiri

Tabel 4. 23 Data respon kecepatan motor pergeseran maju kiri

Jarak (m)	Kecepatan Motor (PWM)							
	Motor 1		Motor 2		Motor 3		Motor 4	
	PWM	RPM	PWM	RPM	PWM	RPM	PWM	RPM
2	1850	5200	1850	5200	1570	1040	1570	1040
1.5	1800	4457	1800	4457	1570	1040	1570	1040
1	1670	2526	1670	2526	1590	1337	1590	1337
0.5	1570	1040	1570	1040	1570	1040	1570	1040

Tabel 4. 24 Data servo gerak mundur kanan

Data Servo Mundur Kanan		
Servo	PWM	Sudut
1	2300	90
2	1150	45
3	780	90
4	1500	0

Pada pengujian pergeseran maju kiri, respon pergerakan platform adalah bergerak mundur kanan. Pada pergerakan mundur kanan motor yang akan bergerak dalam kondisi moving adalah motor 1 dan motor 2, sedangkan motor 3 dan 4 dalam kondisi kecepatan stand-by. Pada data yang didapatkan dari simulasi matlab didapatkan respon bahwa semakin jauh jarak pergeseran platform terhadap setpoint kecepatan motor akan semakin tinggi.

c. Pergeseran Mundur Kanan

Tabel 4. 25 Data respon kecepatan motor pergeseran mundur kanan

Jarak (m)	Kecepatan Motor (PWM)							
	Motor 1		Motor 2		Motor 3		Motor 4	
	PWM	RPM	PWM	RPM	PWM	RPM	PWM	RPM
2	1570	1040	1570	1040	1850	5200	1850	5200
1.5	1570	1040	1570	1040	1800	4457	1800	4457
1	1590	1337	1590	1337	1670	2526	1670	2526
0.5	1570	1040	1570	1040	1570	1040	1570	1040

Tabel 4. 26 Data servo gerak maju kiri

Data Servo Maju Kiri		
Servo	PWM	Sudut
1	1500	0
2	800	90
3	1150	45
4	2300	90

Pada pengujian pergeseran mundur kanan, respon pergerakan platform adalah bergerak maju kiri. Pada pergerakan maju kiri motor yang akan bergerak dalam kondisi moving adalah motor 3 dan motor 4, sedangkan motor 1 dan 2 dalam kondisi kecepatan stand-by. Pada data yang didapatkan dari simulasi matlab didapatkan respon bahwa semakin jauh jarak pergeseran platform terhadap setpoint kecepatan motor akan semakin tinggi.

#### d. Pergeseran Mundur Kiri

Tabel 4. 27 Data respon kecepatan motor pergeseran mundur kiri

Jarak (m)	Kecepatan Motor (PWM)							
	Motor 1		Motor 2		Motor 3		Motor 4	
	PWM	RPM	PWM	RPM	PWM	RPM	PWM	RPM
2	1850	5200	1850	5200	1570	1040	1570	1040
1.5	1800	4457	1800	4457	1570	1040	1570	1040
1	1670	2526	1670	2526	1590	1337	1590	1337
0.5	1570	1040	1570	1040	1570	1040	1570	1040

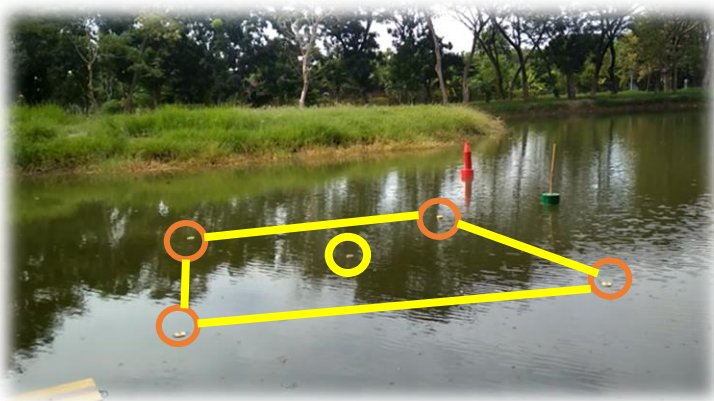
Tabel 4. 28 Data servo gerak maju kanan

Data Servo Maju Kanan		
Servo	PWM	Sudut
1	1900	45
2	800	90
3	1500	0
4	2300	90

Pada pengujian pergeseran mundur kiri, respon pergerakan platform adalah bergerak maju kanan. Pada pergerakan maju kanan motor yang akan bergerak dalam kondisi moving adalah motor 1 dan motor 2, sedangkan motor 3 dan 4 dalam kondisi kecepatan stand-by. Pada data yang didapatkan dari simulasi matlab didapatkan respon bahwa semakin jauh jarak pergeseran platform terhadap setpoint kecepatan motor akan semakin tinggi.

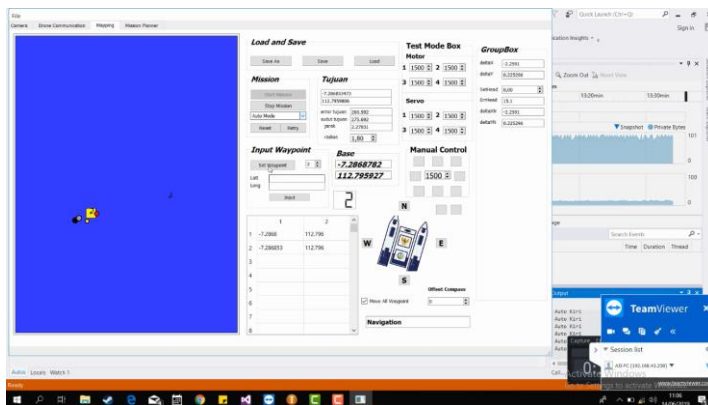
#### 4.9. Validasi Alat

Validasi merupakan pencocokan antara hasil yang didapat pada uji coba dengan hasil pengukuran pada standar yang ada. Pada hal ini validasi yang dilakukan adalah membandingkan hasil pengukuran yang dilakukan pada kondisi ujicoba sesungguhnya di danau dengan menggunakan buoy yang dibandingkan dengan sistem monitoring yang dibuat untuk mengetahui respon pergerakan platform dari jarak jauh atau didarat. Dari bacaan respon dari sistem monitoring tersebut akan dapat diketahui apakah nilai eror dari Model platform semi-submersible sudah memenuhi target yang sudah ditentukan di Bab 3. Berikut merupakan perbandingan penampakan data yang terdapat di danau dengan yang terdapat di sistem monitoring platform.



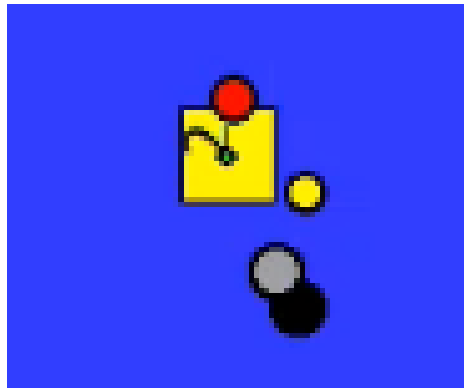
Gambar 4. 50 Penampakan lintasan ujicoba

Gambar diatas merupakan penampakan kondisi real arena atau lintasan untuk uji coba Dynamic Positioning System pada kondisi real dimana lintasan diberi tanda berupa buoy. Buoy yang diberi tanda lingkaran warna jingga adalah sebagai penanda batasan zona stand-by dari Model platform semisubmersible. Sedangkan buoy yang diberi tanda lingkaran warna kuning adalah sebagai titik tujuan atau refrensi point dari platform untuk mempertahankan posisinya pada titik tersebut.

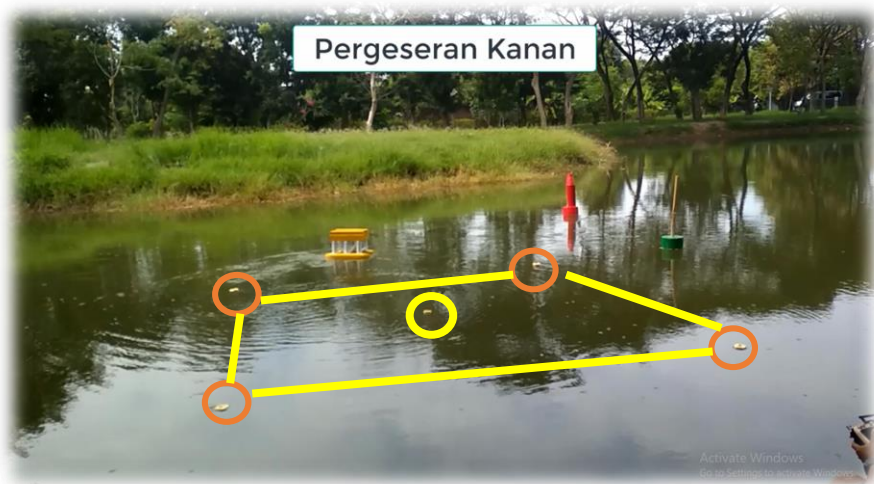


Gambar 4. 51 Penampakan sistem monitoring platform dari darat

Gambar diatas merupakan penampakan sistem monitoring dari pergerakan Dynamic Positioning System Model platform Semi-submersible yang dimonitoring dari darat menggunakan remote control device. Sistem monitoring ini bertujuan untuk mengetahui respon pergerakan platform, nilai data GPS tujuan tau set point platform, eror sudut dari heading platform dan nilai eror jarak posisi platform terhadap set point yang sudah ditentukan. Dari sistem monitoring ini kemudian dapat diketahui apakah nilai eror respon pergerakan platform sudah memenuhi target yaitu maksimal 15 derajat untuk eror heading dan 0.5 m untuk nilai eror posisi platform terhadap titik tujuan.



Gambar 4. 52 Monitoring Pergerakan Platform



Gambar 4. 53 Kondisi Nyata Pergerakan Platform

Gambar diatas merupakan penampakan sistem monitoring pergerakan platform. Dimana platform ditandai dengan titik lingkaran berwarna merah sedangkan refrensi point di darat ditandai dengan titik lingkaran warna hitam dan kotak kuning merupakan penanda zona stand-by pada sistem DPS platform yang dimana pada kondisi nyata ditandai dengan buoy yang dilingkari warna jingga. Gambar diatas menunjukkan platform pada kondisi pergeseran kanan.

<b>Tujuan</b>	
	-7.286853473
	112.7959806
error tujuan	260.592
sudut tujuan	275.692
jarak	2.27031
radius	1,80

Gambar 4. 54 Monitoring Error Jarak Platform Terhadap Titik Tujuan

Dari kondisi pergeseran platform ke kanan dapat diketahui melalui sistem monitoring dari gambar di atas di mana ditunjukkan posisi tujuan platform atau set point platform serta nilai error sudut terhadap tujuan dan jarak platform ke tujuan. Pada gambar di atas nilai jarak platform terhadap tujuan adalah 2.27031 meter. Selain dapat diketahui nilai error jarak, pada sistem monitoring ini juga dapat diketahui nilai error sudut heading platform dari set heading yang sudah ditentukan melalui gambar di bawah ini.

<b>GroupBox</b>	
deltaX	-2.2591
deltaY	0.225296
SetHead	0,00
ErrHead	15.1
deltaXN	-2.2591
deltaYN	0.225296

Gambar 4. 55 Monitoring Error Sudut Platform Terhadap Heading

Dari sistem monitoring ini pada data hasil uji coba dapat diketahui secara langsung nilai error platform baik itu nilai error jarak ke tujuan ataupun nilai error sudut terhadap set heading. Dari hasil uji coba dan validasi ini diketahui bahwa nilai error jarak platform masih belum memenuhi target yang ditentukan yaitu kurang dari 0.5 m. Hal ini ditunjukkan dari respon pergerakan pada pergeseran arah Surge, pergeseran kanan, serta pergeseran maju kanan yang nilai jarak erornya masih lebih dari 0.5 m. Sedangkan untuk sudut heading platform nilai erornya sudah memenuhi dimana headingnya tidak lebih dari 15 derajat.

*“Halaman Ini Sengaja Dikosongkan”*



## **BAB V. KESIMPULAN DAN SARAN**

### **5.1. Kesimpulan**

Kesimpulan yang didapatkan dari perancangan *Dynamic Positioning System* pada Model *Platform Semi-submersible* adalah sebagai berikut :

1. Perancangan *Dynamic Positioning System* pada Model *Platform Semi-submersible* dengan menggunakan *logika Fuzzy*, menggunakan metode *Differential Global Positioning System* dapat meningkatkan keakuratan positioning pada platform yang erornya menjadi kurang dari 0.5 m.
2. Pada pengujian pergeseran platform, respon pergerakan platform sesuai dengan yang dirancang namun terdapat beberapa nilai eror posisi yang lebih dari 0.5m, yaitu pada respon gerak *Surgee*, respon gerak mundur kiri, dan respon gerak kanan *sway* dengan rata-rata nilai eror 0.55m. Sedangkan untuk respon pergerakan *Yaw* nilai eror sudut sudah dapat terpenuhi yaitu dengan rata-rata 8.3°.
3. Respon kecepatan motor *thruster* terhadap jarak pergeseran adalah semakin jauh jarak pergeseran kecepatan motor *thruster* akan semakin tinggi. Kecepatan motor akan maksimal pada jarak pergeseran 2m atau lebih yaitu 5200 RPM dan 0 – 1040 RPM pada jarak pergeseran kurang dari 1m.
4. Respon waktu tercepat pergerakan Model platform *Semi-submersible* terhadap 4 skenario pergeseran adalah 23 detik, yaitu pada respon pergerakan Mundur terhadap pergeseran Model platform ke arah Maju dengan rata-respon waktu adalah 36 detik.

### **5.2. Saran**

Saran dari penulis terhadap pembuatan sistem kendali DPS pada Model *Platform Semi-submersible* adalah :

1. Dilakukan perhitungan untuk pemodelan matematis dari desain serta respon pergerakan untuk didapatkan persamaan yang lebih akurat pada sistem kontrol DPS.
2. Pada penelitian selanjutnya dapat menggunakan sensor GPS dengan akurasi yang lebih tinggi serta ditambahkan sensor lain seperti sensor angin, dan sensor *proximity* lainnya untuk meningkatkan akurasi respon posisi platform.

*“Halaman Ini Sengaja Dikosongkan”*

## DAFTAR PUSTAKA

- [1] M. Yamamoto dan C. K. Morooka, "Dynamic Positioning System of Semi-Submersible Platform Using Fuzzy Control," 2005.
- [2] J. Lopes-Cortijo, A. S. Dungal, R. R. van Dijk dan S. Matos, "DP FPSO - A Fully Positioned FPSO for Ultra Deep Waters," *International Offshore and Polar Engineering Conference*, 2003.
- [3] C. S. Chas dan R. Ferreiro, "Introduction To Ship Dynamic Positioning System," *Journal of Maritime Research*, 2008.
- [4] Y. Bai dan Q. Bai, "Instalation and Vessel," dalam *Subsea Engineering Handbook*, 2010, pp. 139-158.
- [5] M. J. Morgan, "PPC Books Division," *Dynamic Positioning of Offshore Vessels*, 1978.
- [6] ABS (American Bureau of Shipping), "Classification Notation," dalam *Guide for Dynamic Positioning System*, 2013, p. 1.
- [7] J. Halkyard, "Floating Offshore Platform Design," dalam *Handbook of Offshore Engineering*, 2005, pp. 419-661.
- [8] C. Bo, Y. Zhiyong, L. Yong, L. Xiaojian dan L. Chunfang, "A new Type of Anti-heave Semi-submersible Drilling Platform," 2017.
- [9] S. I. Baniela, P. L. Varela dan E. M. Rodriguez, "Concept And Operation Mode of The Advance Electronic Control System of The Azimuth Propellers In Tugs," 2005.
- [10] STMicroelectronics Group, UM1472 User Manual, Januari 2014 penyunt., 2014.
- [11] E. Damaranda, "S.T. Thesis," *Pembuatan Purwarupa Tes Model Untuk Menguji Sistem Autopilot Pada USV.*, 2017.
- [12] H. Khoirudin, "S.T. Thesis," *Capstone Desain Kapal Tanpa Awak Sebagai Media Survei Tingkat Kualitas Air di Suatu Perairan*, 2018.
- [13] Ublox, "NEO-M8 - Datasheet". *NEO-M8 u-blox M8 concurrent GNSS modules*.
- [14] Phishrobot.com, "User Manual CMPS-11," *CMPS-11 Tilt Compansated Compass Module*, 2015.
- [15] Fatjay, "Data Sheet," *waterproof 2838 350KV brushless outrunner motor with 60mm propeller 2.4KG thrust for underwater thruster ROV e-foil RC boats*, 2018.
- [16] www.aliexpress.com, "HC-12 Wireless Serial Port Communication Module," p. 10, 2018.
- [17] L. Reznik, *Fuzzy Controllers*, Newnes, 1997.
- [18] D. S. Chandrasekaran, "Drill Ships and Basic of Drilling," dalam *Ocean Structure and Materials*, 2005.

*“Halaman Ini Sengaja Dikosongkan”*

## LAMPIRAN A PROGRAM ARDUINO BASE REFERENCE

```
#include <string.h>

float lattitude; //float tdk di dalam loop ??
float longitude;

int count = 0;
int penandaGPS = 2;
int penandaGPS2 = 0;
char buffer_gps[100];
int t = 0;

int bluetooth = 3;
int blue;
char kirim[40];

void setup() {
    // put your setup code here, to run once:
    Serial1.begin(9600); //bluetooth
    Serial.begin(9600);
    Serial2.begin(9600); // ke STM
    Serial3.begin(9600); //GPS
}
char aaa[20];
char bbb[20];
char ccc[20];
```

78

```
char ddd[20];
```

```
char eee[20];
```

```
char fff[20];
```

```
char ggg[20];
```

```
char hhh[20];
```

```
char iii2[20];
```

```
char jjj7[20];
```

```
char kkk[20];
```

```
int la = 0;
```

```
int lo = 0;
```

```
char su;
```

```
void loop() {
```

```
    if(Serial1.available(>0)
```

```
    {
```

```
        bluetooth = (int)Serial1.read();
```

```
        bluetooth = bluetooth - 48;
```

```
    }
```

```
    if(Serial3.available(>0)
```

```
    {
```

```
        su = Serial3.read();
```

```
    if(penandaGPS == 2)
```

```
    {
```

```
if(su == 'R')
{
    penandaGPS = 3;
    //Serial.print("[R]");
}
else{
    penandaGPS = 2;
}
}
else if(penandaGPS == 3)
{
    if(su == 'M')
    {
        penandaGPS = 4;
        //Serial.print("[M]");
    }
    else{
        penandaGPS = 2;
    }
}
else if(penandaGPS == 4)
{
    if(su == 'C')
    {
        penandaGPS = 5;
        //Serial.print("[C]");
    }
}
```

```
    }  
    else{  
        penandaGPS = 2;  
    }  
}  
else if(penandaGPS == 5)  
{  
    if(su != '$')  
    {  
        if(penandaGPS2 == 0){  
            if( su != ',')  
            {  
  
            }  
            else{  
                //Serial.print("{}");  
                penandaGPS2 = 1;  
            }  
        }  
        else if(penandaGPS2 == 1){  
            if( su != ',')  
            {  
  
            }  
            else{  
                //Serial.print("{}");
```



```
penandaGPS2 = 2;
}
}
else if(penandaGPS2 == 2){
    if( su != ',')
    {

    }
    else{
        //Serial.print("{}");
        penandaGPS2 = 3;
    }
}
else if(penandaGPS2 == 3){
    if( su != ',')
    {
        //Serial.print(la);
        ccc[la++] = su;
        //Serial.print("|");

    }
    else{
        //Serial.print("{}");
        penandaGPS2 = 4;
    }
}
}
```

```
else if(penandaGPS2 == 4){
    if( su != ',')
    {

    }
    else{
        //Serial.print("{}");
        penandaGPS2 = 5;
    }
}
else if(penandaGPS2 == 5){
    if( su != ',')
    {
        //Serial.print(lo);
        eee[lo++] = su;
        //Serial.print(" |");

    }
    else{
        //Serial.print("{}");
        penandaGPS2 = 6;
    }
}
else if(penandaGPS2 == 6){
    if( su != ',')
    {
```

```
    }  
    else{  
        //Serial.print("{}");  
        penandaGPS2 = 7;  
    }  
}  
}  
}  
else  
{  
    penandaGPS = 2;  
    penandaGPS2 = 0;  
    t=0;  
    la = 0;  
    lo = 0;  
  
    sprintf(kirim,"AB%d",(int)bluetooth);  
    Serial2.print(kirim);  
    Serial.print(kirim);  
    Serial2.print(ccc);  
    Serial.print(ccc);  
    sprintf(kirim,"");  
    Serial2.print(kirim);  
    Serial.print(kirim);  
    Serial2.print(eee);  
    Serial.print(eee);
```

```
    sprintf(kirim,"C");  
    Serial2.print(kirim);  
    Serial.println(kirim);  
  }  
}  
}  
}
```

## LAMPIRAN B

### PROGRAM STM KONTROL PLATFORM

```

/* CATATAN PEMBACAAN REMOT!!
 * Nilai pembacaan pulsa remot antara 2000 sampai 4000
 *
 *          INPUT                OUTPUT
 *      E9   (TIM1_CH1)  ->      A6   (TIM3_CH1)
 *      E11  (TIM1_CH2)  ->      A7   (TIM3_CH2)
 *      E13  (TIM1_CH3)  ->      B0   (TIM3_CH3)
 *      E14  (TIM1_CH4)  ->      B1   (TIM3_CH4)
 *      C6   (TIM8_CH1)  ->      D12  (TIM4_CH1)
 *      C7   (TIM8_CH2)  ->      D13  (TIM4_CH2)
 *      C8   (TIM8_CH3)  ->      D14  (TIM4_CH3)
 *      C9   (TIM8_CH4)  ->      D15  (TIM4_CH4)
 *
 *      USART 1 => send data to PC           => Tx B6 Rx A10
 *      USART 2 => get data GPS              => Tx A2 Rx A3
 *      USART 3 => get compass               => Tx D8 Rx D9
 *      UART 4  => send data with HC-12      => Tx A0 Rx A1
 *
 *
 *
 *
 */
#include "stm32f4xx.h"
#include "stm32f4xx_conf.h"
#include "stdio.h"
#include "delay.h"
#include "stdlib.h"
#include "lcd_biasa.h"
#include "math.h"
#include "filter.h"
#include "string.h"
#include "inisialisasi_timer.h"
#include "inisialisasi_io.h"
#include "inisialisasi_USART.h"
#include "tm_stm32f4_i2c.h"

float delta_x=0;
float delta_X=0;
float delta_y=0;
float delta_Y=0;

double longitude_DGPS;
double lattitude_DGPS;
double latt_Tujuan;
double long_Tujuan;
double BaseAwalLong;

```

```
double BaseAwallLatt;
double PlatformAwallLong;
double PlatformAwallLatt;

double sudutNav;
double sudutTujuanNav;
double errorSudutNav;
double erorSudutHeading;
float PI = 3.14;

double fuzzidelta_X;
double kondisidelta_X;
double fuzzidelta_Y;
double kondisidelta_Y;
double fuzzyHeading;
double kondisiErrorHeading;

// Definisi variabel fungsi
void monitoringData ();
void ambildata ();
void ambildata2 ();
void ambil_data_pc ();
void ambil_data_arduino ();

int Motor1 = 1500;
int Motor2 = 1500;
int Motor3 = 1500;
int Motor4 = 1500;
int Servo1 = 1500;
int Servo2 = 1500;
int Servo3 = 1500;
int Servo4 = 1500;

//kondisi servo1
float serong1 = 1900;
float lurus1 = 1500;
float sampling1 = 2300;

//kondisi servo2
float serong2 = 1150;
float lurus2 = 1500;
float sampling2 = 800;

//kondisi servo3
float serong3 = 1140;
float lurus3 = 1500;
float sampling3 = 780;

//kondisi servo4
float serong4 = 1900;
float lurus4 = 1500;
```

```

float sampling4 = 2300;

char tampil[32];
char kirimString[100];
int counter_drone=0;

#define out1 TIM3->CCR1
#define out2 TIM3->CCR2
#define out3 TIM3->CCR3
#define out4 TIM3->CCR4
#define out5 TIM4->CCR1
#define out6 TIM4->CCR2
#define out7 TIM4->CCR3
#define out8 TIM4->CCR4

signed long int data1[2];
signed long int data2[2];
signed long int data3[2];
signed long int data4[2];
signed long int data5[2];
signed long int data6[2];
signed long int data7[2];
signed long int data8[2];

signed long int lebar_pulsa1=3000;
signed long int lebar_pulsa2=3000;
signed long int lebar_pulsa3=3000;
signed long int lebar_pulsa4=3000;
signed long int lebar_pulsa5=3000;
signed long int lebar_pulsa6=3000;
signed long int lebar_pulsa7=3000;
signed long int lebar_pulsa8=3000;

#define ADDRESS_KOMPAS 0xC0;
#define transmit 0x00;
#define receive 0x01;
#define tombol_lcd GPIO_ReadInputDataBit(GPIOD,GPIO_PIN_3)
int counter_lcd = 0;

long int pewaktu;
long int pewaktu_i2c;
long int pewaktu1ms;
long int pewaktu100ms;
long int pewaktu_kontrol;
int state_remot=0;
char data_siap;

//Data ke Remote
#define falling 1
#define rising 0

```

```

#define kondisi_polaritas1_rising ((TIM1->CCER)>>1 & 1)==0
#define kondisi_polaritas1_falling ((TIM1->CCER)>>1 & 1)==1
#define kondisi_polaritas2_rising ((TIM1->CCER)>>5 & 1)==0
#define kondisi_polaritas2_falling ((TIM1->CCER)>>5 & 1)==1
#define kondisi_polaritas3_rising ((TIM1->CCER)>>9 & 1)==0
#define kondisi_polaritas3_falling ((TIM1->CCER)>>9 & 1)==1
#define kondisi_polaritas4_rising ((TIM1->CCER)>>13 & 1)==0
#define kondisi_polaritas4_falling ((TIM1->CCER)>>13 & 1)==1
#define kondisi_polaritas5_rising ((TIM8->CCER)>>1 & 1)==0
#define kondisi_polaritas5_falling ((TIM8->CCER)>>1 & 1)==1
#define kondisi_polaritas6_rising ((TIM8->CCER)>>5 & 1)==0
#define kondisi_polaritas6_falling ((TIM8->CCER)>>5 & 1)==1
#define kondisi_polaritas7_rising ((TIM8->CCER)>>9 & 1)==0
#define kondisi_polaritas7_falling ((TIM8->CCER)>>9 & 1)==1
#define kondisi_polaritas8_rising ((TIM8->CCER)>>13 & 1)==0
#define kondisi_polaritas8_falling ((TIM8->CCER)>>13 & 1)==1

#define polaritas1_falling (TIM1->CCER|=(1<<1))
#define polaritas1_rising (TIM1->CCER &= ~(1<<1))
#define polaritas2_falling (TIM1->CCER|=(1<<5))
#define polaritas2_rising (TIM1->CCER &= ~(1<<5))
#define polaritas3_falling (TIM1->CCER|=(1<<9))
#define polaritas3_rising (TIM1->CCER &= ~(1<<9))
#define polaritas4_falling (TIM1->CCER|=(1<<13))
#define polaritas4_rising (TIM1->CCER &= ~(1<<13))
#define polaritas5_falling (TIM8->CCER|=(1<<1))
#define polaritas5_rising (TIM8->CCER &= ~(1<<1))
#define polaritas6_falling (TIM8->CCER|=(1<<5))
#define polaritas6_rising (TIM8->CCER &= ~(1<<5))
#define polaritas7_falling (TIM8->CCER|=(1<<9))
#define polaritas7_rising (TIM8->CCER &= ~(1<<9))
#define polaritas8_falling (TIM8->CCER|=(1<<13))
#define polaritas8_rising (TIM8->CCER &= ~(1<<13))

#define periode_tim1 40000-1
#define periode_tim8 40000-1

#define SERVOKIRI out3
#define SERVOKANAN out4
#define SERVO_DRONE out5=out6
#define MOTORKIRI out7
#define MOTORKANAN out8

float sudut16;
float heading = 0;
float error_heading = 0;
float error_heading_sebelum = 0;
float P_head = 18.0;
float P_heads = 35.0;
float I_head = 1.1;

```



```

float I_heads = 3.1;
float D_head = 8.3;
float D_heads = 13.7;

float pid_motor;
float pid_motor_serong;

int gerak = 0;
int move = 0;

void lcd(char kolom, char baris, char*string)
{
    lcd_gotoxy(kolom,baris);
    lcd_puts(string);
}

signed long int scale_motor;
signed long int scale_motor_mundur;
int batas_bawah_servo=800;
int batas_atas_servo=2200;

void olah_pwm_remot();
void test_servo();
void test_motor();

void test_servo()
{
    error_heading = sudut16 - heading;
    if(error_heading>180.0)
    {
        error_heading = error_heading - 360.0;
    }
    else if(error_heading<(-180.0))
    {
        error_heading = 360.0 - error_heading;
    }
    pid_motor = P_head*error_heading -
D_head*error_heading_sebelum + I_head*error_heading_sebelum;
    pid_motor_serong = P_heads*error_heading -
D_heads*error_heading_sebelum + I_heads*error_heading_sebelum;
    error_heading_sebelum = error_heading;
    //out1 motor belakang kanan
    //out2 motor belakang kiri
    //out3 servo belakang kiri
    //out4 servo belakang kanan

    //out5 motor depan kiri
    //out6 motor depan kanan
    //out7 servo depan kiri
    //out8 servo depan kanan

```

```

        if(lebar_pulsa4<3070 && lebar_pulsa4>2950 &&
lebar_pulsa1 > 2900 && lebar_pulsa1 < 3100 && lebar_pulsa2 >
2900 && lebar_pulsa2 < 3100)
    {
        gerak = 0;
        move = 1;
    }
    else {
        gerak = 1;
        if(move == 1){
            heading = sudut16;
            move = 0;
        }
    }
if(lebar_pulsa4<3070 && lebar_pulsa4>2950){
    if(lebar_pulsa2 < 2900) //
    {
        if(lebar_pulsa1 < 2900) //maju kiri
        {
            out3 = 1810;
            out4 = 1140;
            out7 = 800;
            out8 = 2150;
            if(error_heading < 0)
            {
                out2 = 1670 + abs(pid_motor);
                out5 = 1490;
                out6 = 1490;
            }
            else{
                out6 = 1670 +
abs(pid_motor_serong);
                out2 = 1490;
                out5 = 1490;
            }
            if(abs(lebar_pulsa2-3000) >
abs(lebar_pulsa1-3000))
            {
                out1 = 4520 - lebar_pulsa2;
            }
            else{
                out1 = 4520 - lebar_pulsa1;
            }
        }
        else if(lebar_pulsa1 > 3100) //maju kanan
        {
            out3 = 1810;
            out4 = 1140;
            out7 = 800;
            out8 = 2150;

```

```

        if(error_heading < 0)
        {
            out5 = 1670 +
abs(pid_motor_serong);
            out1 = 1490;
            out6 = 1490;
        }
        else{
            out1 = 1670 + abs(pid_motor);
            out6 = 1490;
            out5 = 1490;
        }
        if(abs(lebar_pulsa2-3000) >
abs(lebar_pulsa1-3000))
        {
            out2 = 4520 - lebar_pulsa2;
        }
        else{
            out2 = lebar_pulsa1 - 1480;
        }
    }
    else{ //maju
        out3 = 1810;
        out4 = 1140;
        out7 = 800;
        out8 = 2150;

        out1 = 4520 - lebar_pulsa2;
        out2 = 4520 - lebar_pulsa2;

        if(error_heading < 0)
        {
            out5 = 1590 + abs(pid_motor);
            out6 = 1490;
        }
        else{
            out6 = 1590 + abs(pid_motor);
            out5 = 1490;
        }
    }
}
else if(lebar_pulsa2 > 3100) //mundur
{
    //out5 = out6 = lebar_pulsa2 - 1480;
    if(lebar_pulsa1 < 2900) //mundur kiri
    {
        out3 = 2300;
        out4 = 780;
        out7 = 1150;
        out8 = 1810;
    }
}

```

```

        if(error_heading < 0)
        {
            out1 = 1670 +
abs(pid_motor_serong);
            out3 = 1490;
            out5 = 1490;
        }
        else{
            out5 = 1670 + abs(pid_motor);
            out3 = 1490;
            out1 = 1490;
        }

        if(abs(lebar_pulsa2-3000) >
abs(lebar_pulsa1-3000))
        {
            out6 = lebar_pulsa2 - 1480;
        }
        else{
            out6 = 4520 - lebar_pulsa1;
        }
    }
    else if(lebar_pulsa1 > 3100) //mundur kanan
    {
        if(error_heading < 0)
        {
            out2 = 1490;
            out6 = 1670 + abs(pid_motor);
            out1 = 1490;
        }
        else{
            out2 = 1670 +
abs(pid_motor_serong);
            out1 = 1490;
            out6 = 1490;
        }

        if(abs(lebar_pulsa2-3000) >
abs(lebar_pulsa1-3000))
        {
            out5 = 4520 - lebar_pulsa2;
        }
        else{
            out5 = lebar_pulsa1 - 1480;
        }
    }
    else{ //mundur
        out7 = 1150;
        out8 = 1810;

        out3 = 2300;
    }
}

```

```

out4 = 750;

out5 = lebar_pulsa2 - 1480;
out6 = lebar_pulsa2 - 1480;

if(error_heading < 0)
{
    out1 = 1590 + abs(pid_motor);
    out2 = 1490;
}
else{
    out2 = 1590 + abs(pid_motor);
    out1 = 1490;
}
}
else{
if(lebar_pulsa1 < 2900) // kiri
{
    out4 = 1140;
    out8 = 1810;

    out3 = 1500;
    out7 = 1500;

    out1 = 4520 - lebar_pulsa1;
    out6 = 4520 - lebar_pulsa1;

if(error_heading < 0)
{
    out2 = 1590 + abs(pid_motor);
    out5 = 1490;
}
else{
    out5 = 1590 + abs(pid_motor);
    out2 = 1490;
}
}
else if(lebar_pulsa1 > 3100) //kanan
{
    out4 = 1500;
    out8 = 1500;

    out3 = 1810;
    out7 = 1150;

    out2 = lebar_pulsa1 - 1480;
    out5 = lebar_pulsa1 - 1480;

if(error_heading < 0)
{

```

```

        out6 = 1590 + abs(pid_motor);
        out1 = 1490;
    }
    else{
        out1 = 1590 + abs(pid_motor);
        out6 = 1490;
    }
}
else{ // diam
    out3 = 1810;
    out4 = 1140;
    out7 = 1150;
    out8 = 1810;
    out5 = out6 = out1 = out2 = 1490;
}
}

else if(lebar_pulsa4<=2950) //rotate kiri
{
    out3 = 2300;
    out4 = 1140;
    out7 = 1150;
    out8 = 2150;
    out2 = out6 = 4520 - lebar_pulsa4;
    out1 = out5 = 1490;
}
else if(lebar_pulsa4>=3070) // rotate kanan
{
    out3 = 1810;
    out4 = 780;
    out7 = 800;
    out8 = 1810;
    out1 = out5 = lebar_pulsa4 - 1480;
    out2 = out6 = 1490;
}

}

void test_motor()
{
    out3 = out4 = out7 = out8 = 1500;
    out1 = out2 = out5 = out6 = 4520 - lebar_pulsa2;
}

void olah_pwm_remot()
{
    //out1 = 1500;
    //out2 = 1500;
    //out7 = 1500;
    //out8 = 1500;

```

```

//out3 = out4 = out7 = out8 = 4500 - lebar_pulsa3;
//out3 = out4 = out7 = out8 = 1500;

//SERVO BELAKANG KIRI
//out3 = 1500; //1810//1950
//SERVO BELAKANG KANAN
//out4 = 1500; //1000
//SERVO DEPAN KIRI
//out7 = 1500; //1100
//SERVO DEPAN KANAN
//out8 = 1500; //1950//1850
//out1 = out2 = out5 = out6 = 4520 - lebar_pulsa2;

//MAJU LEBAR PULSA3 2100
//out5 = out6 = 4520 - lebar_pulsa2;

//out1 motor belakang kanan
//out2 motor belakang kiri
//out3 servo belakang kiri
//out4 servo belakang kanan

//out5 motor depan kiri
//out6 motor depan kanan
//out7 servo depan kiri
//out8 servo depan kanan

}

void TIM1_CC_IRQHandler()
{
    if(TIM_GetFlagStatus(TIM1,TIM_FLAG_CC1)!=0)
    {
        if(kondisi_polaritas1_rising)
        {
            data1[0]=TIM_GetCapture1(TIM1);
            polaritas1_falling;
            data_siap=0;
        }

        else if(kondisi_polaritas1_falling)
        {
            data1[1]=TIM_GetCapture1(TIM1);
            polaritas1_rising;

            if(data1[0]<data1[1]) lebar_pulsa1=data1[1]-
data1[0];
            else lebar_pulsa1=data1[1]+periode_tim1-
data1[0];
            data_siap=1;
        }
    }
}

```

```

        TIM_ClearFlag(TIM1, TIM_FLAG_CC1);
    }

    if(TIM_GetFlagStatus(TIM1, TIM_FLAG_CC2) !=0)
    {
        if(kondisi_polaritas2_rising)
        {
            data2[0]=TIM_GetCapture2(TIM1);
            polaritas2_falling;
            data_siap=0;
        }

        else if(kondisi_polaritas2_falling)
        {
            data2[1]=TIM_GetCapture2(TIM1);
            polaritas2_rising;

            if(data2[0]<data2[1]) lebar_pulsa2=data2[1]-
data2[0];
            else lebar_pulsa2=data2[1]+periode_tim1-
data2[0];
            data_siap=1;
        }
        TIM_ClearFlag(TIM1, TIM_FLAG_CC2);
    }

    if(TIM_GetFlagStatus(TIM1, TIM_FLAG_CC3) !=0)
    {
        if(kondisi_polaritas3_rising)
        {
            data3[0]=TIM_GetCapture3(TIM1);
            polaritas3_falling;
            data_siap=0;
        }

        else if(kondisi_polaritas3_falling)
        {
            data3[1]=TIM_GetCapture3(TIM1);
            polaritas3_rising;

            if(data3[0]<data3[1]) lebar_pulsa3=data3[1]-
data3[0];
            else lebar_pulsa3=data3[1]+periode_tim1-
data3[0];
            data_siap=1;
        }
        TIM_ClearFlag(TIM1, TIM_FLAG_CC3);
    }

    if(TIM_GetFlagStatus(TIM1, TIM_FLAG_CC4) !=0)
    {

```



```

        if(kondisi_polaritas4_rising)
        {
            data4[0]=TIM_GetCapture4(TIM1);
            polaritas4_falling;
            data_siap=0;
        }

        else if(kondisi_polaritas4_falling)
        {
            data4[1]=TIM_GetCapture4(TIM1);
            polaritas4_rising;

            if(data4[0]<data4[1]) lebar_pulsa4=data4[1]-
data4[0];
            else lebar_pulsa4=data4[1]+periode_tim1-
data4[0];

            data_siap=1;
        }
        TIM_ClearFlag(TIM1,TIM_FLAG_CC4);
    }

}

void TIM8_CC_IRQHandler()
{
    if(TIM_GetFlagStatus(TIM8,TIM_FLAG_CC1)!=0)
    {
        if(kondisi_polaritas5_rising)
        {
            data5[0]=TIM_GetCapture1(TIM8);
            polaritas5_falling;
            data_siap=0;
        }
        else if(kondisi_polaritas5_falling)
        {
            data5[1]=TIM_GetCapture1(TIM8);
            polaritas5_rising;

            if(data5[0]<data5[1]) lebar_pulsa5=data5[1]-
data5[0];
            else lebar_pulsa5=data5[1]+periode_tim8-
data5[0];

            data_siap=1;
        }
        TIM_ClearFlag(TIM8,TIM_FLAG_CC1);
    }
    if(TIM_GetFlagStatus(TIM8,TIM_FLAG_CC2)!=0)
    {
        if(kondisi_polaritas6_rising)
        {
            data6[0]=TIM_GetCapture2(TIM8);

```

```

        polaritas6_falling;
        data_siap=0;
    }
    else if(kondisi_polaritas6_falling)
    {
        data6[1]=TIM_GetCapture2(TIM8);
        polaritas6_rising;

        if(data6[0]<data6[1]) lebar_pulsa6=data6[1]-
data6[0];
        else lebar_pulsa6=data6[1]+periode_tim8-
data6[0];
        data_siap=1;
    }
    TIM_ClearFlag(TIM8,TIM_FLAG_CC2);
}

if(TIM_GetFlagStatus(TIM8,TIM_FLAG_CC3)!=0)
{
    if(kondisi_polaritas7_rising)
    {
        data7[0]=TIM_GetCapture3(TIM8);
        polaritas7_falling;
        data_siap=0;
    }
    else if(kondisi_polaritas7_falling)
    {
        data7[1]=TIM_GetCapture3(TIM8);
        polaritas7_rising;

        if(data7[0]<data7[1]) lebar_pulsa7=data7[1]-
data7[0];
        else lebar_pulsa7=data7[1]+periode_tim8-
data7[0];
        data_siap=1;
    }
    TIM_ClearFlag(TIM8,TIM_FLAG_CC3);
}
if(TIM_GetFlagStatus(TIM8,TIM_FLAG_CC4)!=0)
{
    if(kondisi_polaritas8_rising)
    {
        data8[0]=TIM_GetCapture4(TIM8);
        polaritas8_falling;
        data_siap=0;
    }
    else if(kondisi_polaritas8_falling)
    {
        data8[1]=TIM_GetCapture4(TIM8);
        polaritas8_rising;
    }
}

```

```

        if(data8[0]<data8[1])lebar_pulsa8=data8[1]-
data8[0];
        else lebar_pulsa8=data8[1]+periode_tim8-
data8[0];
        data_siap=1;
    }
    TIM_ClearFlag(TIM8,TIM_FLAG_CC4);
}

void baca_kondisi_remot()
{
    if(lebar_pulsa8>3200)
    {
        state_remot=3; //Mode Otomatis
    }
    else
    {
        if(lebar_pulsa7 > 3200){
            state_remot = 0; //Mode Test Motor Servo 1500
        }
        else if(lebar_pulsa7 < 2800)
        {
            state_remot = 2; //Mode Manual
        }
        else{
            state_remot = 1; //Mode Test Servo
        }
    }
}

void TIM6_DAC_IRQHandler()
{
    TIM_ClearFlag(TIM6,TIM_FLAG_Update);
    pewaktulms++;
    pewaktu_i2c++;

    baca_kondisi_remot();
    if(state_remot==0)olah_pwm_remot();

    if(pewaktulms%100==0)
    {
        pewaktu_kontrol++;
        pewaktu++;
    }
}

double lat_combo;
double long_combo;

char data_uart4;

```

```

int penanda_kompas = 0;
unsigned char data_kompas_serial[3];

void UART4_IRQHandler ()
{
    if (USART_GetITStatus (UART4, USART_IT_RXNE))
    {
        if (penanda_kompas==0)
        {
            data_kompas_serial[0]=USART_ReceiveData (UART4);
            penanda_kompas=1;
        }
        else if (penanda_kompas==1)
        {
            data_kompas_serial[1]=USART_ReceiveData (UART4);
            penanda_kompas=0;
        }
    }
    USART_ClearITPendingBit (UART4, USART_IT_RXNE);
}

void minta_data_kompas (char data)
{
    USART_SendData (UART4, data);
    while (USART_GetFlagStatus (UART4, USART_FLAG_TXE) == RESET);
}

char data_usart2;
char penanda_gps=0;
char buffer_gps[100];

int t2=0;
int t1=0;
char ambil_usart2[50];

void USART2_IRQHandler (void)
{
    if (USART_GetITStatus (USART2, USART_IT_RXNE))
    {
        data_usart2=USART_ReceiveData (USART2);
        ambil_data_arduino ();
    }
    USART_ClearITPendingBit (USART2, USART_IT_RXNE);
}

int penanda_usart2 = 0;
long lattitude;
float latt_base;
float latt_base_sebelum;
float latt_base_murni_sebelum;

```

```

float error_latt_base;
long longitude;
float long_base;
float long_base_sebelum;
float long_base_murni_sebelum;
float error_long_base;
int bluetooth;

void ambil_data_arduino()
{
    if(penanda_usart2==0)
    {
        if(data_usart2=='A'){penanda_usart2=1;}
        else {penanda_usart2 = 0;}
    }
    else if(penanda_usart2==1)
    {
        if(data_usart2=='B'){penanda_usart2=2;}
        else {penanda_usart2 = 0;}
    }
    else if(penanda_usart2==2)
    {
        if(data_usart2!='C')
        {
            ambil_usart2[t2++]= data_usart2;
        }
        else
        {
            sscanf(ambil_usart2,"%d,%li,%li",&bluetooth,&lattitude,&lon
gitude);

            penanda_usart2=0;
            if(bluetooth>11)
            {
                bluetooth = 9;
            }
            t2=0;
            //memset(ambil_usart2,'0',50);
        }
    }
}

char a[20];
char b[20];
char c[20];
char d[20];
char e[20];
char f[20];
char g[20];
char h[20];
char i2[20];

```

```
char j7[20];
char k[20];

void ambildata ()
{
    int i=1;
    int j=0;

    while(buffer_gps[i]!='\n')
    {
        a[j++]=buffer_gps[i++];
    }

    i++;j=0;
    while(buffer_gps[i]!='\n')
    {
        b[j++]=buffer_gps[i++];
    }

    i++;j=0;
    while(buffer_gps[i]!='\n')
    {
        c[j++]=buffer_gps[i++];
    }

    i++;j=0;
    while(buffer_gps[i]!='\n')
    {
        d[j++]=buffer_gps[i++];
    }

    i++;j=0;
    while(buffer_gps[i]!='\n')
    {
        e[j++]=buffer_gps[i++];
    }

    i++;j=0;
    while(buffer_gps[i]!='\n')
    {
        f[j++]=buffer_gps[i++];
    }

    i++;j=0;
    while(buffer_gps[i]!='\n')
    {
        g[j++]=buffer_gps[i++];
    }

    i++;j=0;
    while(buffer_gps[i]!='\n')
```

```

    {
        h[j++]=buffer_gps[i++];
    }

    i++;j=0;
    while(buffer_gps[i]!='\0')
    {
        i2[j++]=buffer_gps[i++];
    }

    i++;j=0;
    while(buffer_gps[i]!='\0')
    {
        j7[j++]=buffer_gps[i++];
    }

    i++;j=0;
    while(buffer_gps[i]!='\0')
    {
        k[j++]=buffer_gps[i++];
        if(buffer_gps[i]==0) break;
    }

    i++;j=0;
}

double lattDouble;
double longDouble;
double lattDecimal=-7.277865;
double longDecimal=112.765421;
int ratusan,puluhan,satuan;
int ratusan2,puluhan2,satuan2;
double menit_ke_sudut,menit_ke_sudut2;
void konvertStringGps ()
{
    latt_base_sebelum = latt_base;
    long_base_sebelum = long_base;
    if(lattitude > 70000000 && lattitude < 75000000 &&
longitude > 1100000000 && longitude < 1150000000)
    {
        latt_base = lattitude/10000000.0*(-1);
        long_base = longitude/10000000.0;
    }
    if(latt_base != latt_base_sebelum)
    {
        latt_base_murni_sebelum = latt_base_sebelum;
        error_latt_base = latt_base - latt_base_sebelum;
    }
    if(long_base != long_base_sebelum)

```

```

    {
        long_base_murni_sebelum = long_base_sebelum;
        error_long_base = long_base - long_base_sebelum;
    }
// lattDouble=atof(c);
// longDouble=atof(e);
//
// //konversi dms ke desimal lattitude
// ratusan=lattDouble/100;
// puluhan=(lattDouble-ratusan*100)/10;
// satuan=(lattDouble-ratusan*100-puluhan*10);
// menit_ke_sudut=(lattDouble-(ratusan*100))/60;
// lattDecimal=(ratusan+menit_ke_sudut)*(-1);
//
// //konversi dms ke desimal longitude
// ratusan2=longDouble/100;
// puluhan2=(longDouble-ratusan2*100)/10;
// satuan2=(longDouble-ratusan2*100-puluhan2*10);
// menit_ke_sudut2=(longDouble-(ratusan2*100))/60;
// longDecimal=(ratusan2+menit_ke_sudut2);
}

```

```

unsigned int data_kontrol[10];

```

```

char ambil_usart1[50];

```

```

char data_usart1;

```

```

int penanda_usart1=0;

```

```

int hitung_usart1 = 0;

```

```

void USART1_IRQHandler(void)

```

```

{
    if(USART_GetITStatus(USART1, USART_IT_RXNE))
    {
        data_usart1=USART_ReceiveData(USART1);
        //hitung_usart1++;
        ambil_data_pc();
    }
    USART_ClearITPendingBit(USART1, USART_IT_RXNE);
}

```

```

void ambil_data_pc()

```

```

{
    if(penanda_usart1==0)
    {
        if(data_usart1=='A'){penanda_usart1=1;}
        else {penanda_usart1 = 0;}
    }
    else if(penanda_usart1==1)
    {

```



```

        if(data_usart1=='B'){penanda_usart1=2;}
        else {penanda_usart1 = 0;}
    }
else if(penanda_usart1==2)
{
    if(data_usart1!='C')
    {
        ambil_usart1[t1++]= data_usart1;
    }
    else
    {

        sscanf(ambil_usart1,"%4d,%4d,%4d,%4d,%4d,%4d,%4d,%4dC",&Motor1,&Motor2,&Motor3,&Motor4,&Servo1,&Servo2,&Servo3,&Servo4);
        penanda_usart1=0;
        t1=0;
    }
}
}

void buffer_data_pc()
{
    int i=1;
    int j=0;

    while(buffer_gps[i]!='\n')
    {
        a[j++]=buffer_gps[i++];
    }

    i++;j=0;
    while(buffer_gps[i]!='\n')
    {
        b[j++]=buffer_gps[i++];
    }

    i++;j=0;
    while(buffer_gps[i]!='\n')
    {
        c[j++]=buffer_gps[i++];
    }

    i++;j=0;
    while(buffer_gps[i]!='\n')
    {
        d[j++]=buffer_gps[i++];
    }
}

int data_usart3;

```

```

unsigned char data_kompas[2];
char buffer_gps2[100];
int t3 = 0;

int penanda_gps2 = 0;

void USART3_IRQHandler(void)
{
    if(USART_GetITStatus(USART3, USART_IT_RXNE))
    {
        data_usart3=USART_ReceiveData(USART3);
        if(penanda_gps2==0)
        {
            if(data_usart3=='G')penanda_gps2=1;
            else penanda_gps2 = 0;
        }
        else if(penanda_gps2==1)
        {

if(data_usart3=='P' || data_usart3=='N')penanda_gps2=2;
            else penanda_gps2 = 0;
        }
        else if(penanda_gps2==2)
        {
            if(data_usart3=='R')penanda_gps2=3;
            else penanda_gps2 = 0;
        }
        else if(penanda_gps2==3)
        {
            if(data_usart3=='M')penanda_gps2=4;
            else penanda_gps2 = 0;
        }
        else if(penanda_gps2==4)
        {
            if(data_usart3=='C')penanda_gps2=5;
            else penanda_gps2 = 0;
        }
        else if(penanda_gps2==5)
        {
            if(data_usart3!='$')
            {
                buffer_gps2[t3++]= data_usart3;
            }
            else
            {
                penanda_gps2=0;
                t3=0;
                ambildata2();
            }
        }
    }
}

```

```
        USART_ClearITPendingBit(USART2, USART_IT_RXNE);
    }
    char aaa[20];
    char bbb[20];
    char ccc[20];
    char ddd[20];
    char eee[20];
    char fff[20];
    char ggg[20];
    char hhh[20];
    char iii2[20];
    char jjj7[20];
    char kkk[20];

    void ambildata2()
    {
        int iii=1;
        int jjj=0;

        while(buffer_gps2[iii]!='\n')
        {
            aaa[jjj++]=buffer_gps2[iii++];
        }

        iii++;jjj=0;
        while(buffer_gps2[iii]!='\n')
        {
            bbb[jjj++]=buffer_gps2[iii++];
        }

        iii++;jjj=0;
        while(buffer_gps2[iii]!='\n')
        {
            ccc[jjj++]=buffer_gps2[iii++];
        }

        iii++;jjj=0;
        while(buffer_gps2[iii]!='\n')
        {
            ddd[jjj++]=buffer_gps2[iii++];
        }

        iii++;jjj=0;
        while(buffer_gps2[iii]!='\n')
        {
            eee[jjj++]=buffer_gps2[iii++];
        }

        iii++;jjj=0;
        while(buffer_gps2[iii]!='\n')
        {
```

```

        fff[jjj++]=buffer_gps2[iii++];
    }

    iii++;jjj=0;
    while(buffer_gps2[iii]!='\0')
    {
        ggg[jjj++]=buffer_gps2[iii++];
    }

    iii++;jjj=0;
    while(buffer_gps2[iii]!='\0')
    {
        hhh[jjj++]=buffer_gps2[iii++];
    }

    iii++;jjj=0;
    while(buffer_gps2[iii]!='\0')
    {
        iii2[jjj++]=buffer_gps2[iii++];
    }

    iii++;jjj=0;
    while(buffer_gps2[iii]!='\0')
    {
        jjj7[jjj++]=buffer_gps2[iii++];
    }

    iii++;jjj=0;
    while(buffer_gps2[iii]!='\0')
    {
        kkk[jjj++]=buffer_gps2[iii++];
        if(buffer_gps2[iii]==0) break;
    }

    iii++;jjj=0;

}

double lattDouble2;
double longDouble2;
double lattDecimal2=-7.277865;
double lattDecimal2Sebelum;
double lattDecimal2MurniSebelum;
double errorLatt1;
double longDecimal2=112.765421;
double longDecimal2Sebelum;
double longDecimal2MurniSebelum;
double errorLong1;
int ratusan11,puluhan11,satuan11;
int ratusan22,puluhan22,satuan22;
double menit_ke_sudut11,menit_ke_sudut22;

```

```

void konvertStringGps2()
{
    lattDecimal2Sebelum = lattDecimal2;
    longDecimal2Sebelum = longDecimal2;
    lattDouble2=atof(ccc);
    longDouble2=atof(eee);

    //konversi dms ke desimal lattitude
    ratusan11=lattDouble2/100;
    puluhan11=(lattDouble2-ratusan11*100)/10;
    satuan11=(lattDouble2-ratusan11*100-puluhan11*10);
    menit_ke_sudut11=(lattDouble2-(ratusan11*100))/60;

    lattDecimal2=(ratusan11+menit_ke_sudut11)*(-1);
    if(lattDecimal2 != lattDecimal2Sebelum)
    {
        lattDecimal2MurniSebelum = lattDecimal2Sebelum;
        errorLatt1 = lattDecimal2 - lattDecimal2Sebelum;
    }

    //konversi dms ke desimal longitude
    ratusan22=longDouble2/100;
    puluhan22=(longDouble2-ratusan22*100)/10;
    satuan22=(longDouble2-ratusan22*100-puluhan22*10);
    menit_ke_sudut22=(longDouble2-(ratusan22*100))/60;

    longDecimal2=(ratusan22+menit_ke_sudut22);
    if(longDecimal2 != longDecimal2Sebelum)
    {
        longDecimal2MurniSebelum = longDecimal2Sebelum;
        errorLong1 = longDecimal2 - longDecimal2Sebelum;
    }
}

unsigned int angle16;

int kirimsudut;

void kirim_data_ke_pc(char data)
{
    USART_SendData(USART1,data);
    while(USART_GetFlagStatus(USART1,USART_FLAG_TXE) ==
RESET);
}

int counter=0;

void kirim_ke_pc()
{

```

```

    int a;
    sprintf(kirimString, "A%2.7lfB%3.7lfC%d%2.7fE%3.7fG", fabs (
lat_combo), long_combo, angle16, fabs (latt_base), long_base);
    //Dihilangkan Degrees
    for (a=0;a<=strlen (kirimString);a++)
    {
        USART_SendData (USART1, kirimString[a]);
        while (USART_GetFlagStatus (USART1, USART_FLAG_TXE) ==
RESET);
    }
}

uint8_t compass[4];

void monitoringData (int menu)
{
    switch (menu)
    {
        case 0:
            sprintf (tampil, "1. Tugas Akhir 2019");
            lcd (0, 0, tampil);
            sprintf (tampil, "    Luthfi Halim E55");
            lcd (0, 1, tampil);
            sprintf (tampil, "    Khalif Aji P55");
            lcd (0, 2, tampil);
            sprintf (tampil, "    Wisuda 120 Amin");
            lcd (0, 3, tampil);
            break;

        case 1:
            sprintf (tampil, "2. Motor");
            lcd (0, 0, tampil);
            sprintf (tampil, "    M1:%4d
M2:%4d", (int)Motor1, (int)Motor2);
            lcd (0, 1, tampil);
            sprintf (tampil, "    M3:%4d
M4:%4d", (int)Motor3, (int)Motor4);
            lcd (0, 2, tampil);
            break;

        case 2:
            sprintf (tampil, "3. Servo");
            lcd (0, 0, tampil);
            sprintf (tampil, "    S1:%4d
S2:%4d", (int)Servo1, (int)Servo2);
            lcd (0, 1, tampil);
            sprintf (tampil, "    S3:%4d
S4:%4d", (int)Servo3, (int)Servo4);
            lcd (0, 2, tampil);
            break;

        case 3:
            sprintf (tampil, "4. DATA GPS Compass");
            lcd (0, 0, tampil);

```

```

        sprintf(tampil, " %2.7lf", lat_combo);
        lcd(0,1,tampil);
        sprintf(tampil, " %3.7lf", long_combo);
        lcd(0,2,tampil);
        sprintf(tampil, " %5.1f ", sudut16);
        lcd(0,3,tampil);
        break;

    case 4:
        sprintf(tampil, "5.Output");
        lcd(0,0,tampil);
        sprintf(tampil, "%4d %4d %4d", (int)out1,
(int)out2, (int)out3);
        lcd(0,1,tampil);
        sprintf(tampil, "%4d %4d %4d", (int)out4,
(int)out5, (int)out6);
        lcd(0,2,tampil);
        sprintf(tampil, "%4d %4d", (int)out7,
(int)out8);
        lcd(0,3,tampil);
        break;

    case 5:
        sprintf(tampil, "6.Data Remote");
        lcd(0,0,tampil);
        sprintf(tampil, "%4d %4d
%4d", (int)lebar_pulsa1, (int)lebar_pulsa2, (int)lebar_pulsa3);
        lcd(0,1,tampil);
        sprintf(tampil, "%4d %4d
%4d", (int)lebar_pulsa4, (int)lebar_pulsa5, (int)lebar_pulsa6);
        lcd(0,2,tampil);
        sprintf(tampil, "%4d
%4d", (int)lebar_pulsa7, (int)lebar_pulsa8);
        lcd(0,3,tampil);
        break;

    case 6:
        sprintf(tampil, "7.Test Data PC");
        lcd(0,0,tampil);
        sprintf(tampil, "Usart 1 :
%d", (int)data_usart1);
        lcd(0,1,tampil);
        sprintf(tampil, "Count : %d",
(int)hitung_usart1);
        lcd(0,2,tampil);
        sprintf(tampil, "Moly");
        lcd(0,3,tampil);
        break;

    case 7:

        sprintf(tampil, "8.GPS1%2.7lf", lattDecimal2MurniSebelum);
        lcd(0,0,tampil);
        sprintf(tampil, "
%2.7lf", lattDecimal2);

```

```

        lcd(0,1,tampil);
        sprintf(tampil,"
GPS2%2.7lf",latt_base_murni_sebelum);
        lcd(0,2,tampil);
        sprintf(tampil,"
%2.7lf",latt_base);
        lcd(0,3,tampil);
        break;
    case 8:
        sprintf(tampil,"9.GPS1%3.7lf",longDecimal2MurniSebelum);
        lcd(0,0,tampil);
        sprintf(tampil,"
%3.7lf",longDecimal2);
        lcd(0,1,tampil);
        sprintf(tampil,"
GPS2%3.7lf",long_base_murni_sebelum);
        lcd(0,2,tampil);
        sprintf(tampil,"
%3.7lf",long_base);
        lcd(0,3,tampil);
        break;
    case 9:
        sprintf(tampil,"10. Data dari Arduino");
        lcd(0,0,tampil);
        sprintf(tampil,"lat :
%3.7f", (float)latt_base);
        lcd(0,1,tampil);
        sprintf(tampil,"long:
%3.7f", (float)long_base);
        lcd(0,2,tampil);
        sprintf(tampil,"bluetooth : %4d",
(int)bluetooth);
        lcd(0,3,tampil);
        break;
    case 10:
        sprintf(tampil,"11. Test Sensor");
        lcd(0,0,tampil);
        sprintf(tampil,"Sudut :
%4.1f",sudut16);
        lcd(0,1,tampil);
        sprintf(tampil,"Heading :
%4.1f",heading);
        lcd(0,2,tampil);

        sprintf(tampil,"Err:%4.1fMot:%4.1f",error_heading,pid_moto
r);
        lcd(0,3,tampil);
        break;
    case 11 :

```



```

        sprintf(tampil, "12.GPS1%2.7lf", lattDecimal2MurniSebelum);
        lcd(0,0,tampil);
        sprintf(tampil, "
%2.7lf",lattDecimal2);
        lcd(0,1,tampil);
        sprintf(tampil, "
GPS2%2.7lf",latt_base_murni_sebelum);
        lcd(0,2,tampil);
        sprintf(tampil, "
%2.7lf",latt_base);
        lcd(0,3,tampil);
        break;
    }
}

void servo_motor()
{
    out7 = Servo1;
    out8 = Servo2;
    out5 = Motor1;
    out6 = Motor2;
    out2 = Motor3;
    out1 = Motor4;
    out3 = Servo3;
    out4 = Servo4;
}

void reset_data()
{
    Motor1 = 1500;
    Motor2 = 1500;
    Motor3 = 1500;
    Motor4 = 1500;
    Servo1 = 1500;
    Servo2 = 1500;
    Servo3 = 1500;
    Servo4 = 1500;
}

int main(void)
{
    //Semua Inisiasi tulis disini

    SystemInit();

    RCC_APB2PeriphClockCmd(RCC_APB2Periph_SYSCFG, ENABLE);
//enable interrupt
    RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOA, ENABLE);

```

```

RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOB, ENABLE);
RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOC, ENABLE);
RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOD, ENABLE);
RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOE, ENABLE);
RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOH, ENABLE);
RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOG, ENABLE);

inisialisasi_timer6();
inisialisasi_io();

lcd_init(20);

reset_data();

pwm_motor();
pwm_remot();

inisialisasi_USART2();
inisialisasi_USART1();
inisialisasi_USART3();
inisialisasi_UART4();

out1=1500;
out2=1500;
out3=1500;
out4=1500;
out5=1500;
out6=1500;
out7=1500;
out8=1500;

TIM8->CNT=0;
TIM1->CNT=0;

while(1)
{

    konvertStringGps(); //olah data GPS 1
    konvertStringGps2(); //olah data GPS 2

    //gabungin 2 data GPS
    if(lattDecimal < -7.5 || lattDecimal > -7.0)
    {
        lat_combo = lattDecimal2;
    }
    else if(lattDecimal2 < -7.5 || lattDecimal2 > -7.0)
    {
        lat_combo = lattDecimal;
    }
    else {

```

```

        lat_combo = (lattDecimal + lattDecimal2)/2.0;
    }

    if(longDecimal < 112.0 || longDecimal > 113.0)
    {
        long_combo = longDecimal2;
    }
    else if(longDecimal2 < 112.0 || longDecimal2 >
113.0)
    {
        long_combo = longDecimal;
    }
    else{
        long_combo = (longDecimal + longDecimal2)/2.0;
    }

    minta_data_kompas(19);
    angle16=((data_kompas_serial[0]<<8) +
data_kompas_serial[1]);
    sudut16=angle16/10.0;

    //Algorithm DGPS

    longitude_DGPS = (double)((9 * long_combo + 7 *
(long_base - (BaseAwalLong + PlatformAwalLong))) / 16);
    lattitude_DGPS = (double)((9 * lat_combo + 7 *
(latt_base - (BaseAwalLatt + PlatformAwalLatt))) / 16);

    //Algorithm Input Error Heading

    sudutNav = (atan2(delta_y, delta_x)*(180 / PI));
    sudutTujuanNav = fmod((450.0 - sudutNav), 360.0);
    errorSudutNav = (sudutTujuanNav - sudut16);
    erorSudutHeading = (sudut16 - heading);

    //Algorithm Input GPS

    delta_x=(longitude_DGPS - long_Tujuan)*111320;
    delta_y=(lattitude_DGPS - latt_Tujuan)*110574.6108;

    delta_X=(delta_x * cos(sudutTujuanNav / PI)) -
(delta_y * sin(sudutTujuanNav / PI));
    delta_Y = (delta_y/cos(sudutTujuanNav / PI)) +
(delta_X * tan(sudutTujuanNav / PI));

    // FUZIFIKASI

    // deltaX

    if ((delta_X >= 0.75)&&(delta_X <= 2)){
        fuzzidelta_X = (delta_X-0.75)/(2-0.75);

```

```

        kondisidelta_X = 1; //deltaXP
    }
    else if ((delta_X >= -0.75)&&(delta_X <= -2)){
        fuzzidelta_X = (-0.75-delta_X)/(-2-(-0.75));
        kondisidelta_X = 2; //deltaXN
    }
    else if ((delta_X >= -1.25)&&(delta_X <= 0)){
        fuzzidelta_X = (delta_X-(-1.25))/(1.25);
        kondisidelta_X = 3; //StandbyXP
    }
    else if ((delta_X >= 0)&&(delta_X <= 1.25)){
        fuzzidelta_X = (1.25-delta_X)/(1.25);
        kondisidelta_X = 3; //StandbyXP
    }
}

//deltaY

if ((delta_Y >= 0.75)&&(delta_Y <= 2)){
    fuzzidelta_Y = (delta_Y-0.75)/(2-0.75);
    kondisidelta_Y = 1; // deltaYP
}
else if ((delta_Y >= -0.75)&&(delta_Y <= -2)){
    fuzzidelta_Y = (-0.75-delta_Y)/(-2-(-0.75));
    kondisidelta_Y = 2; // deltaYN
}
else if ((delta_Y >= -1.25)&&(delta_Y <= 0)){
    fuzzidelta_Y = (delta_Y-(-1.25))/(1.25);
    kondisidelta_Y = 3; // StandbyYP
}
else if ((delta_Y >= 0)&&(delta_Y <= 1.25)) {
    fuzzidelta_Y = (1.25-delta_Y)/(1.25);
    kondisidelta_Y = 3; // StandbyYP
}

//erorHeading

if ((erorSudutHeading >= 15)&&(erorSudutHeading <
180)){
    fuzzyHeading = (erorSudutHeading - 15)/(180-
15);
    kondisiErrorHeading = 1; //CW
}
else if ((erorSudutHeading >= -
180)&&(erorSudutHeading <=-15)){
    fuzzyHeading = (-15-erorSudutHeading)/(-15-(-
90));
    kondisiErrorHeading = 2; //CCW
}
else if ((erorSudutHeading >= -
15)&&(erorSudutHeading <= 0)){

```

```

fuzzyHeading = (erorSudutHeading - (-
15))/ (15);
    kondisiErrorHeading = 3; //StandbyHeading
}
else if ((erorSudutHeading >= 0)&&(erorSudutHeading
<= 15)){
    fuzzyHeading = (15-erorSudutHeading)/ (15);
    kondisiErrorHeading = 3; //StandbyHeading
}

// FUZZY RULES

    if ((kondisidelta_X = 3)&&(kondisidelta_Y =
3)&&(kondisiErrorHeading = 3)) {
        Servo1=serong1;
        Servo2=serong2;
        Servo3=serong3;
        Servo4=serong4;
        Motor1=((fuzzyHeading*(1500-
1700)+1700)+(fuzzidelta_X*(1500-1700)+1700)+(fuzzidelta_Y*(1500-
1700)+1700)/3);
        Motor2=((fuzzyHeading*(1500-
1700)+1700)+(fuzzidelta_X*(1500-1700)+1700)+(fuzzidelta_Y*(1500-
1700)+1700)/3);
        Motor3=((fuzzyHeading*(1500-
1700)+1700)+(fuzzidelta_X*(1500-1700)+1700)+(fuzzidelta_Y*(1500-
1700)+1700)/3);
        Motor4=((fuzzyHeading*(1500-
1700)+1700)+(fuzzidelta_X*(1500-1700)+1700)+(fuzzidelta_Y*(1500-
1700)+1700)/3);
    }//stand by
    else if ((kondisidelta_X = 3)&&(kondisidelta_Y =
2)&&(kondisiErrorHeading = 3)) {
        Servo1=lurus1;
        Servo2=sampling2;
        Servo3=lurus3;
        Servo4=sampling4;
        Motor1=(fuzzidelta_Y*(1850-1700)+1700);
        Motor2=((fuzzyHeading*(1500-
1700)+1700)+(fuzzidelta_X*(1500-1700)+1700)/2);
        Motor3=(fuzzidelta_Y*(1850-1700)+1700);
        Motor4=((fuzzyHeading*(1500-
1700)+1700)+(fuzzidelta_X*(1500-1700)+1700)/2);
    }//maju
    else if ((kondisidelta_X = 3)&&(kondisidelta_Y =
1)&&(kondisiErrorHeading = 3)) {
        Servo1=sampling1;
        Servo2=lurus2;
        Servo3=sampling3;
        Servo4=lurus2;

```

```

        Motor1=((fuzzyHeading*(1500-
1700)+1700)+(fuzzidelata_X*(1500-1700)+1700)/2);
        Motor2=((fuzzidelata_Y*(1850-
1700)+1700));
        Motor3=((fuzzyHeading*(1500-
1700)+1700)+(fuzzidelata_X*(1500-1700)+1700)/2);
        Motor4=((fuzzidelata_Y*(1850-
1700)+1700));
    }//mundur
    else if ((kondisidelata_X = 2)&&(kondisidelata_Y =
3)&&(kondisiErrorHeading = 3)) {
        Servo1=samping1;
        Servo2=samping2;
        Servo3=lurus3;
        Servo4=lurus4;
        Motor1=((fuzzidelata_X*(1850-
1700)+1700));
        Motor2=((fuzzidelata_X*(1850-
1700)+1700));
        Motor3=((fuzzyHeading*(1500-
1700)+1700)+(fuzzidelata_Y*(1500-1700)+1700)/2);
        Motor4=((fuzzyHeading*(1500-
1700)+1700)+(fuzzidelata_Y*(1500-1700)+1700)/2);
    }//kanan
    else if ((kondisidelata_X = 1)&&(kondisidelata_Y =
3)&&(kondisiErrorHeading = 3)) {
        Servo1=lurus1;
        Servo2=lurus2;
        Servo3=samping3;
        Servo4=samping4;
        Motor1=((fuzzyHeading*(1500-
1700)+1700)+(fuzzidelata_Y*(1500-1700)+1700)/2);
        Motor2=((fuzzyHeading*(1500-
1700)+1700)+(fuzzidelata_Y*(1500-1700)+1700)/2);
        Motor3=((fuzzidelata_X*(1850-
1700)+1700));
        Motor4=((fuzzidelata_X*(1850-
1700)+1700));
    }//kiri
    else if ((kondisidelata_X = 2)&&(kondisidelata_Y =
2)&&(kondisiErrorHeading = 3)) {
        Servo1=serong1;
        Servo2=samping2;
        Servo3=lurus3;
        Servo4=samping4;
        Motor1=((fuzzidelata_X*(1850-
1700)+1700)+(fuzzidelata_Y*(1850-1700)+1700)/2);
        Motor2=((fuzzidelata_X*(1850-
1700)+1700)+(fuzzidelata_Y*(1850-1700)+1700)/2);
        Motor3=(fuzzyHeading*(1500-1700)+1700);
        Motor4=(fuzzyHeading*(1500-1700)+1700);

```

```

        }//kanan depan
        else if ((kondisidelta_X = 2)&&(kondisidelta_Y =
1)&&(kondisiErrorHeading = 3)) {
            Servo1=samping1;
            Servo2=serong2;
            Servo3=samping3;
            Servo4=lurus4;
            Motor1=((fuzzidelta_X*(1850-
1700)+1700)+(fuzzidelta_Y*(1850-1700)+1700))/2);
            Motor2=((fuzzidelta_X*(1850-
1700)+1700)+(fuzzidelta_Y*(1850-1700)+1700))/2);
            Motor3=(fuzzyHeading*(1500-1700)+1700);
            Motor4=(fuzzyHeading*(1500-1700)+1700);
        }//kanan belakang
        else if ((kondisidelta_X = 1)&&(kondisidelta_Y =
2)&&(kondisiErrorHeading = 3)) {
            Servo1=lurus1;
            Servo2=samping2;
            Servo3=serong3;
            Servo4=samping4;
            Motor1=(fuzzyHeading*(1500-1700)+1700);
            Motor2=(fuzzyHeading*(1500-1700)+1700);
            Motor3=((fuzzidelta_X*(1850-
1700)+1700)+(fuzzidelta_Y*(1850-1700)+1700))/2);
            Motor4=((fuzzidelta_X*(1850-
1700)+1700)+(fuzzidelta_Y*(1850-1700)+1700))/2);
        }//kiri depan
        else if ((kondisidelta_X = 1)&&(kondisidelta_Y =
1)&&(kondisiErrorHeading = 3)) {
            Servo1=samping1;
            Servo2=lurus2;
            Servo3=samping3;
            Servo4=serong4;
            Motor1=(fuzzyHeading*(1500-1700)+1700);
            Motor2=(fuzzyHeading*(1500-1700)+1700);
            Motor3=((fuzzidelta_X*(1850-
1700)+1700)+(fuzzidelta_Y*(1850-1700)+1700))/2);
            Motor4=((fuzzidelta_X*(1850-
1700)+1700)+(fuzzidelta_Y*(1850-1700)+1700))/2);
        }//kiri belakang
        else if ((kondisiErrorHeading = 2)) {
            Servo1=lurus1;
            Servo2=samping2;
            Servo3=samping3;
            Servo4=lurus4;
            Motor1=1500
            Motor2=(fuzzyHeading*(1850-1700)+1700);
            Motor3=(fuzzyHeading*(1850-1700)+1700);
            Motor4=1500
        }//cw
        else if ((kondisiErrorHeading = 1)) {

```

```

        Servo1=sampling1;
        Servo2=lurus2;
        Servo3=lurus3;
        Servo4=sampling4;
        Motor1=(fuzzyHeading*(1850-1700)+1700);
        Motor2=1500;
        Motor3=1500;
        Motor4=(fuzzyHeading*(1850-1700)+1700);
    }//ccw

    baca_kondisi_remot();
    tampilan LCD
    if(tombol_lcd == 1 || lebar_pulsa6 > 3200)//ganti
    {
        lcd_clear();
        counter_lcd++;
        if(counter_lcd>11)
        {
            counter_lcd = 0;
        }
        delay_ms(600);
    }

    switch(state_remot)
    {
        case 0: //manual
            test_motor();
            GPIO_SetBits(GPIOE, GPIO_Pin_5);
            GPIO_ResetBits(GPIOE, GPIO_Pin_4);
            break;
        case 1:
            test_servo();
            GPIO_SetBits(GPIOE, GPIO_Pin_5);
            GPIO_ResetBits(GPIOE, GPIO_Pin_4);
            break;
        case 2:
            olah_pwm_remot();
            //reset_data();
            GPIO_SetBits(GPIOE, GPIO_Pin_5);
            GPIO_ResetBits(GPIOE, GPIO_Pin_4);
            break;
        case 3: //otomatis
            servo_motor();

            GPIO_SetBits(GPIOE, GPIO_Pin_4);
            GPIO_ResetBits(GPIOE, GPIO_Pin_5);
            break;
    }

```



```
LCD      monitoringData(counter_lcd); //menampilkan data ke
        }
        kirim_ke_pc();
    }
```

*“Halaman Ini Sengaja Dikosongkan”*

## LAMPIRAN C

### PROGRAM DESIGN FUZZY DI MATHLAB

```
[System]
Name='Logika Fuzzy DPS.2'
Type='mamdani'
Version=2.0
NumInputs=3
NumOutputs=8
NumRules=11
AndMethod='min'
OrMethod='max'
ImpMethod='min'
AggMethod='max'
DefuzzMethod='centroid'

[Input1]
Name='Delta.X'
Range=[-2 2]
NumMFs=3
MF1='e*-X': 'trimf', [-3 -2 -0.75]
MF2='Stand-by': 'trimf', [-1.25 0 1.2]
MF3='e*X': 'trimf', [0.75 2 3]

[Input2]
Name='Delta.Y'
Range=[-2 2]
NumMFs=3
MF1='e*-Y': 'trimf', [-3 -2 -0.75]
MF2='Stand-by': 'trimf', [-1.25 0 1.25]
MF3='e*Y': 'trimf', [0.75 2 3]

[Input3]
Name='Error.Heading.Kompas'
Range=[-180 180]
NumMFs=3
MF1='e*CCW': 'trimf', [-180 -180 -15]
MF2='Stand-by': 'trimf', [-15 0 15]
MF3='e*CW': 'trimf', [15 180 180]

[Output1]
Name='Servo1'
Range=[1500 2300]
NumMFs=3
MF1='Lurus': 'trimf', [1100 1500 1600]
MF2='Serong': 'trimf', [1600 1900 2200]
MF3='Samping': 'trimf', [2200 2300 2700]

[Output2]
```

```
Name='Servo3'  
Range=[780 1500]  
NumMFs=3  
MF1='Samping':'trimf',[420 780 880]  
MF2='Serong':'trimf',[880 1140 1400]  
MF3='Lurus':'trimf',[1400 1500 1860]
```

```
[Output3]  
Name='Motor3'  
Range=[1500 1850]  
NumMFs=2  
MF1='Stand-by':'trimf',[1400 1500 1700]  
MF2='Moving':'trimf',[1700 1850 2000]
```

```
[Output4]  
Name='Motor1'  
Range=[1500 1850]  
NumMFs=2  
MF1='Stand-by':'trimf',[1400 1500 1700]  
MF2='Moving':'trimf',[1700 1850 2000]
```

```
[Output5]  
Name='Motor2'  
Range=[1500 1850]  
NumMFs=2  
MF1='Stand-by':'trimf',[1400 1500 1700]  
MF2='Moving':'trimf',[1700 1850 2000]
```

```
[Output6]  
Name='Motor4'  
Range=[1500 1850]  
NumMFs=2  
MF1='Stand-by':'trimf',[1400 1500 1700]  
MF2='moving':'trimf',[1700 1850 2000]
```

```
[Output7]  
Name='Servo2'  
Range=[800 1500]  
NumMFs=3  
MF1='Samping':'trimf',[450 800 900]  
MF2='Serong':'trimf',[900 1150 1400]  
MF3='Lurus':'trimf',[1400 1500 1850]
```

```
[Output8]  
Name='Servo4'  
Range=[1500 2300]  
NumMFs=3  
MF1='Lurus':'trimf',[1100 1500 1600]  
MF2='Serong':'trimf',[1600 1900 2200]  
MF3='Samping':'trimf',[2200 2300 2700]
```

[Rules]

2 2 2, 2 2 1 1 1 1 2 2 (1) : 1  
 2 1 2, 1 3 2 2 1 1 1 3 (1) : 1  
 2 3 2, 3 1 1 1 2 2 3 1 (1) : 1  
 1 2 2, 3 3 1 2 2 1 1 1 (1) : 1  
 3 2 2, 1 1 2 1 1 2 3 3 (1) : 1  
 1 1 2, 2 3 1 2 2 1 1 3 (1) : 1  
 1 3 2, 3 1 1 2 2 1 2 1 (1) : 1  
 3 1 2, 1 2 2 1 1 2 1 3 (1) : 1  
 3 3 2, 3 1 2 1 1 2 3 2 (1) : 1  
 2 2 1, 1 1 2 1 2 1 1 1 (1) : 1  
 2 2 3, 3 3 1 2 1 2 3 3 (1) : 1

*“Halaman Ini Sengaja Dikosongkan”*

## LAMPIRAN D SPECIFIKASI

### 1. Spesifikasi Motor Brushless

**Specifications:**

KV: 350rpm/V

Configuration: 12N/14P

Max power: 300W

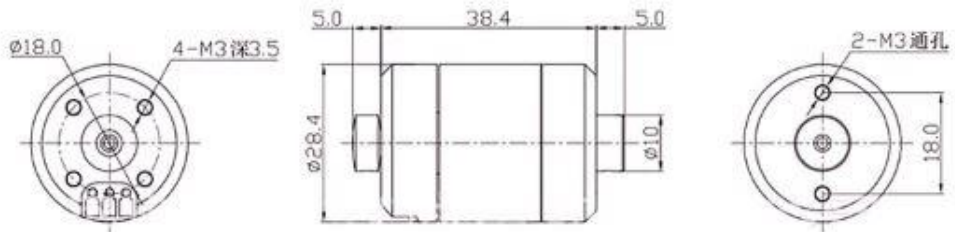
Max current: 15A

Voltage: 3-6S

Weight: 60g

Prop: 60mm diameter, 3-blade CW/CCW

**MOTOR OUTLINE DRAWING (外形图):**



**MOTOR PERFORMANCE DATA (性能参数):**

MODEL	KV (rpm/V)	Voltage (V)	No Load Current (A)	No Load Speed (rpm)	Max Current (A)	Max Power (W)	Lipo Cell	Weight (g) Approx
F2838	350	11.1V	0.25	3950	7.5	83	3-6S	50
		14.8V	0.30	5200	10.0	150		

**Test Data:**

Voltage: 12V, max current: 6A, max thrust: 1.1Kg;

Voltage: 16V, max current: 8A, max thrust: 1.5Kg;

Voltage: 24V, max current: 12A, max thrust: 2.4Kg;

*“Halaman Ini Sengaja Dikosongkan”*



## PROFIL PENULIS



Khalif Aji Puspito, lahir di Kota Palembang, Sumatra Selatan pada tanggal 6 September 1996. Penulis merupakan anak kedua dari tiga bersaudara. Penulis menempuh pendidikan pada tingkat dasar di TK Bayangkara Probolinggo, kemudian melanjutkan pendidikan sekolah dasar di SD Tembokrejo 2 Pasuruan, SMPN 1 Pasuruan, dan pendidikan SMA di SMAN 1 Pasuruan. Setelah lulus dari bangku SMA, penulis diterima dan melanjutkan pendidikan perguruan tinggi di Departemen Teknik Sistem Perkapalan, Fakultas Teknologi Kelautan, Institut Teknologi Sepuluh Nopember.

Selama menempuh pendidikan di Departemen Teknik Sistem Perkapalan ITS, penulis turut aktif di beberapa kegiatan organisasi kampus. Di tahun 2016/2017 penulis menjadi salah satu staf METIC (*Marine Teknologi and Innovation Club*) Himasiskal ITS. ditahun yang sama penulis juga menjadi bagian dari tim riset robot kapal tanpa awak yaitu Tim Barunastra ITS sebagai anggota divisi Mekanik. Ditahun 2017/2018 penulis diberi amanah untuk menjadi *General Manager* Tim Barunastra ITS. Bersama Tim Barunastra ITS, penulis berhasil mendapatkan beberapa gelar penghargaan juara baik Nasional maupun Internasional diantaranya, Juara 1 Nasional KKCTBN PPNS 2016, Juara 1, 2, dan *Best Design* di Deconbotion Undip 2017, Juara 4 *International RoboBoat Competition* 2017 di Florida, USA, Serta Juara 1 *International RoboBoat Competition* 2018 di Florida, USA, yang diadakan oleh Robonation.

E-mail : khalifaji10@gmail.com

Kontak : +6281328800265