



TUGAS AKHIR-ME184834

RANCANG BANGUN DYNAMIC POSITIONING SYSTEM PADA MODEL  
SUPPLY VESSEL MENGGUNAKAN SENSOR ULTRASONIK DENGAN  
KONTROL ARDUINO BERBASIS PID

ALFIUS CHRISTIAN KATILIK  
NRP 0421154000003

Dosen Pembimbing  
Indra Ranu Kusuma, S.T., M.Sc.  
Ir. Agoes Santoso, M.Sc., M.phil.

Departemen Teknik Sistem Perkapalan  
Fakultas Teknologi Kelautan  
Institut Teknologi Sepuluh Nopember  
Surabaya  
2019

“ Halaman ini sengaja dikosongkan”



UNDERGRADUATE THESIS

DESIGN OF DYNAMIC POSITIONING SYSTEM'S ON SUPPLY VESSEL MODEL  
USING ULTRASONIC SENSOR WITH THE PID BASED ARDUINO CONTROL

ALFIUS CHRISTIAN KATILIK  
NRP 0421154000003

Supervisor  
Indra Ranu Kusuma, S.T., M.Sc.  
Ir. Agoes Santoso, M.Sc., M.phil.

DEPARTEMENT OF MARINE ENGINEERING  
FACULTY OF MARINE TECHNOLOGY  
SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY  
SURABAYA  
2019

“ Halaman ini sengaja dikosongkan”

LEMBAR PENGESAHAN

RANCANG BANGUN DYNAMIC POSITIONING SYSTEM PADA  
MODEL SUPPLY VESSEL MENGGUNAKAN SENSOR ULTRASONIK  
DENGAN KONTROL ARDUINO BERBASIS PID

TUGAS AKHIR

Diajukan Guna 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 Teknik Kelautan  
Institut Teknologi Sepuluh Nopember

Oleh

ALFIUS CHRISTIAN KATILIK  
NRP 0421154000003

Disetujui oleh Dosen Pembimbing Tugas Akhir :

Indra Ranu Kusuma, S.T., M.Sc.  
NIP: 1979 0327 2003 12 1001

(  )

Ir. Agoes Santoso, M.Sc., M.phil.  
NIP : 1968 0928 1991 02 1001

(  )

Surabaya  
Agustus, 2019

“ Halaman ini sengaja dikosongkan”

## LEMBAR PENGESAHAN

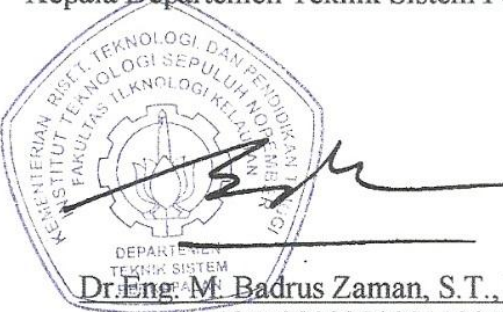
# RANCANG BANGUN DYNAMIC POSITIONING SYSTEM PADA MODEL SUPPLY VESSEL MENGGUNAKAN SENSOR ULTRASONIK DENGAN KONTROL ARDUINO BERBASIS PID

## 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 :  
**ALFIUS CHRISTIAN KATILIK**  
**NRP 0421154000003**

Disetujui oleh :  
Kepala Departemen Teknik Sistem Perkapalan



**Dr. Eng. M. Badrus Zaman, S.T., M.T.**  
NIP. 197708022008011007

“ Halaman ini sengaja dikosongkan”



# RANCANG BANGUN *DYNAMIC POSITIONING SYSTEM* PADA MODEL *SUPPLY VESSEL* MENGGUNAKAN SENSOR ULTRASONIK DENGAN KONTROL ARDUINO BERBASIS PID

Nama : Alfius Christian Katilik  
NRP : 0421154000003  
Jurusan : Teknik Sistem Perkapalan  
Dosen Pembimbing : Indra Ranu Kusuma, S.T., M.Sc.  
: Ir. Agoes Santoso, M.Sc., M.phil.

## ABSTRAK

Rancang Bangun Dynamic Positioning System (DPS) pada model Supply Vessel menggunakan kontrol Arduino berbasis PID merupakan sistem kendali otomatis untuk mempertahankan posisi dan arah hadap kapal selalu tetap pada titik dan arah tertentu menggunakan sistem propulsi. Komponen DPS pada tugas akhir ini, terdiri dari : sistem referensi posisi dan arah menggunakan sensor ultrasonik dan sensor *gyro accelerometer*, sistem kontrol berupa mikrokontroler Arduino untuk mengolah data berbasis PID, sistem penggerak berupa motor *thruster* berjumlah 2 buah dibagian buritan dan haluan kapal , serta sistem pembangkit daya menggunakan baterai. Sistem kontrol akan mengendalikan respon keluaran dari sistem penggerak. Input Sistem kontrol berupa pembacaan nilai besaran jarak dan sudut dari sistem referensi yang kemudian akan dibandingkan dengan nilai *set point* (masukan yang diinginkan). Nilai *set point* divariasikan berdasarkan jarak kapal terhadap model rig dan sudut normal kapal sebesar  $0^0$ . Sistem kontrol akan membandingkan kedua nilai input. Selisih kedua nilai tersebut akan menjadi nilai kesalahan (*error*). Sistem kontrol mengolah data dan mengirimkan perintah berbasis PID untuk mengeliminir nilai error yang dijalankan oleh sistem penggerak. Pengujian dilakukan untuk mengetahui respon kapal dalam mempertahankan posisi dan sudut arah di titik tertentu mencakup gerakan *Sway* dan *Yaw* dengan cara memberikan gaya dorong eksternal berupa arus air pada kapal. Hasil yang didapatkan dari pengujian adalah kapal mampu merepresentasikan respon untuk mempertahankan posisinya pada titik dan arah tertentu. Berdasarkan pengujian , nilai rata rata simpangan jarak (*error*) sebesar 0.06 m pada bagian haluan , 0.03 m pada bagian buritan dari *set point* dan respon waktu rata rata mencapai *set point* sebesar 9 detik . Besar putaran rata rata motor bervariasi yaitu, putaran terendah sebesar 358 rpm dan putaran tertinggi sebesar 416 rpm. Nilai koefisien PID berdasarkan metode Zieger Nichols adalah  $K_p=1.2$ ,  $K_i=0.2$  ,  $K_d= 3.5$ .

## KATA KUNCI :

Dynamic Positioning Sytem, Supply Vessel, Kontrol PID

“ Halaman ini sengaja dikosongkan”

## DESIGN OF DYNAMIC POSITIONING SYSTEM ON SUPPLY VESSEL MODEL USING ULTRASONIC SENSOR WITH THE PID BASED ARDUINO CONTROL

Name : Alfius Christian Katilik  
NRP : 0421154000003  
Departement : Teknik Sistem Perkapalan  
Supervisor : Indra Ranu Kusuma, S.T., M.Sc.  
: Ir. Agoes Santoso, M.Sc., M.phil.

### ABSTRACT

Design of Dynamic Positioning System (DPS) on Supply Vessel model using a PID-based control is an automatic control system to maintain the position and heading of the vessel always remain at a certain point and direction using propulsion system. The DPS component consists of a position and heading reference system using an ultrasonic sensor and gyro accelerometer sensor, a control system using Arduino microcontroller for processing PID-based data, propulsion system using 2 motor thruster on the stern and bow of the vessel, and the power system using batteries. The control system will control the output response of the propulsion system. The input of the control system is a value of the measured distance and angle from the reference system that will then be compared by the value of the setpoint (desire input). The value of the setpoint is varied based on the distance of the vessel from the rig model and the normal angle of the vessel by  $0^{\circ}$ . The control system compares both input values, the difference between the values will be an error value. The Control system processes data and sends PID-based commands to eliminate error values to the propulsion system. Testing was conducted to find out the ship's response to maintain the position and angle at a certain point encompassing the Sway and Yaw movements by providing an external thrust force on the ship in the form of current. The result is that the vessel is able to represent the response to maintain its position at a certain point and direction. Based on the test, the distance deviation rate (error) is 0.06 m on the bow, 0.03 m on the stern from the setpoint and the average time response to reaches a setpoint is 9 seconds. The Average rotation of the motor is varied. The lowest rotation is 358 rpm and the highest rotation is 416 rpm. The value of a PID coefficient based on Zieger Nichols method is  $K_p = 1.2$ ,  $K_i = 0.2$ ,  $K_d = 3.5$ .

### KEY WORD :

Dynamic Positioning System, Supply Vessel, PID Control

“ Halaman ini sengaja dikosongkan”

## KATA PENGANTAR

Puji Tuhan dan mengucapkan syukur kepada Tuhan Yesus karena atas berkat dan karunia Nya, penulis dapat menyelesaikan Tugas Akhir ini dengan judul “RANCANG BANGUN DYNAMIC POSITIONING SYSTEM PADA MODEL SUPPLY VESSEL MENGGUNAKAN SENSOR ULTRASONIK DENGAN KONTROL ARDUINO BERBASIS PID” sebagai salah satu syarat kelulusan program studi sarjana di Departemen Teknik Sistem Perkapalan, Fakultas Teknologi Kelautan, Institut Teknologi Sepuluh Nopember Surabaya.

Dalam proses penyelesaian Tugas Akhir, tidak lepas dari bantuan dan dukungan dari berbagai pihak. Oleh karena itu, penulis menyampaikan rasa terima kasih kepada pihak-pihak yang terkait di bawah ini, yaitu:

1. Bapak Frans Hoknis Katilik dan Ibu Joice Tasiyam sebagai orang tua yang selalu memberikan dukungan doa, dan motivasi hingga Tugas Akhir ini selesai. Abner dan Abraham selaku adik dari penulis yang menjadi dorongan dan motivasi bagi penulis untuk menyelesaikan program studi sarjana di Departemen Teknik Sistem Perkapalan. Segenap keluarga besar dari penulis yang selalu memberikan dukungan berupa Doa, motivasi, maupun materil.
2. Bapak Indra Ranu Kusuma, S.T., M.Sc. dan bapak Ir. Agoes Santoso, M.Sc., M.phil. Selaku dosen pembimbing penulis yang telah bersedia membimbing dan memberikan arahan kepada penulis dalam menyelesaikan Tugas Akhir ini.
3. Bapak Dr. Eng. Dhimas Widhi Handani, S.T., M.Sc. sebagai dosen wali yang selalu memberikan arahan dan motivasi selama penulis menempuh pendidikan di departemen ini.
4. Bapak Dr. Eng Muhammad Badrus Zaman, ST., MT. selaku kepala Departemen Teknik Sistem Perkapalan, FTK, ITS.
5. Keluarga besar UKM Robotika khususnya Tim Barunastra ITS, selaku teman-teman dekat penulis yang selalu bersedia memberikan bantuan kepada penulis untuk menyelesaikan tugas akhir baik dalam bentuk dukungan motivasi, waktu dan sumber daya.
6. Keluarga besar Salvage 15 khususnya teman teman *Laboratorium Marine Electrical and Automation System* sebagai teman seperjuangan penulis bersama dalam mengembangkan diri selama berkuliah di ITS.
7. Kepada pihak yang tidak bisa disebutkan satu per satu yang telah membantu dalam menyelesaikan Tugas Akhir ini, kiranya Tuhan balas berkat.

Penulis menyadari bahwa penelitian yang dilakukan dalam 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, 30 Juli 2019

Alfius Ch Katilik

“ Halaman ini sengaja dikosongkan”

## DAFTAR ISI

LEMBAR PENGESAHAN.....	v
LEMBAR PENGESAHAN.....	vii
<b>ABSTRAK.....</b>	<b>ix</b>
<b>ABSTRACT .....</b>	<b>xi</b>
<b>KATA PENGANTAR .....</b>	<b>xiii</b>
<b>DAFTAR ISI.....</b>	<b>xv</b>
<b>DAFTAR GAMBAR .....</b>	<b>xvii</b>
<b>DAFTAR TABEL.....</b>	<b>xix</b>
<b>BAB I.....</b>	<b>1</b>
<b>1. 1. Latar Belakang .....</b>	<b>1</b>
<b>1. 2. Rumusan Masalah.....</b>	<b>2</b>
<b>1. 3. Batasan Masalah.....</b>	<b>2</b>
<b>1. 4. Tujuan .....</b>	<b>2</b>
<b>BAB II.....</b>	<b>3</b>
<b>2. 1. Dynamic Positioning.....</b>	<b>3</b>
<b>2. 2. Classification Notation .....</b>	<b>14</b>
<b>2. 3. Supply Vessel .....</b>	<b>15</b>
<b>2. 4. Sensor Ultrasonik .....</b>	<b>16</b>
<b>2. 5. Mikrokontroler Arduino .....</b>	<b>17</b>
<b>2. 6. Gyro- Accelerometer .....</b>	<b>23</b>
<b>2. 7. Brushless DC Motor .....</b>	<b>24</b>
<b>2. 8. Electronic Speed Controller .....</b>	<b>24</b>
<b>2. 9. Kontrol PID.....</b>	<b>25</b>
<b>2. 10. Metode Zieger -Nichols .....</b>	<b>27</b>
<b>BAB III.....</b>	<b>31</b>
<b>3. 1. Perumusan Masalah.....</b>	<b>31</b>
<b>3. 2. Studi Literatur .....</b>	<b>31</b>
<b>3. 3. Desain dan Perancangan Alat .....</b>	<b>32</b>
<b>3. 4. Desain dan Rancangan Komponen Sistem Mekanik .....</b>	<b>32</b>

3. 5.	Pembuatan Alat .....	37
3. 6.	Pemograman .....	37
3. 7.	Percobaan Alat.....	37
3. 8.	Analisan Data Percobaan dan Pembahasan.....	38
3. 9.	Kesimpulan dan Saran.....	38
3. 10.	Penyusunan Laporan .....	38
3. 11.	Flow Chart .....	39
<b>BAB IV</b>	.....	<b>41</b>
4. 1.	Gambaran Umum.....	41
4. 2.	Respon yang diharapkan .....	42
4. 3.	Data Model Kapal.....	42
4. 4.	Perancangan Sistem Elektronik.....	44
4. 5.	Pemasangan Komponen.....	51
4. 6.	Logika Pemograman .....	53
4. 7.	Metode Pengujian .....	74
4. 9.	Mencari nilai PWM minimal.....	78
4. 10.	Hasil Pengujian .....	79
<b>BAB V</b>	.....	<b>85</b>
<b>DAFTAR PUSTAKA</b>	.....	<b>87</b>
<b>LAMPIRAN A</b>	.....	<b>89</b>
<b>LAMPIRAN B</b>	.....	<b>95</b>
<b>PROFIL PENULIS</b>	.....	<b>101</b>



## DAFTAR GAMBAR

Gambar 2. 1. Metode Pengeboran Bangunan Lepas Pantai .....	4
Gambar 2. 2. Enam Derajat Kebebasan Kapal .....	5
Gambar 2. 3. Derajat Kebebasan Dynamic Positioning Supply Vessel .....	5
Gambar 2. 4 Light Taut Wire .....	6
Gambar 2. 5. Ultra / Super Short baseline ( USBL /SSBL) .....	7
Gambar 2. 6. Short baseline (SBL) .....	8
Gambar 2. 7. Long Baseline (LBL).....	8
Gambar 2. 8. RadaScan Pada Kapal.....	9
Gambar 2. 9 FanBeam Laser.....	10
Gambar 2. 10. Network DGPS configuration .....	11
Gambar 2. 11. Diagram Blok Sistem Kontrol .....	12
Gambar 2. 12 Supply Vessel .....	15
Gambar 2. 13 SRF HC-04 .....	16
Gambar 2. 14 Prinsip Kerja SRF.....	16
Gambar 2. 15 Mikrokontroler Arduino Uno .....	17
Gambar 2. 16 Bagian Bagian Arduino .....	18
Gambar 2. 17 Sinyal Analog .....	20
Gambar 2. 18. Sinyal Digital.....	21
Gambar 2. 19 Grafik PWM .....	22
Gambar 2. 20 Gyro-Accelerometer .....	23
Gambar 2. 21. Bagian Motor.....	24
Gambar 2. 22. Rangkaian Motor .....	24
Gambar 2. 23 Fungsi PID.....	25
Gambar 2. 24. Skema Kontroler PID .....	26
Gambar 2. 25 Ilustrasi Respon Kontroller .....	26
Gambar 2. 26 Diagram Blok PID .....	27
Gambar 2. 27 Fungsi Alih PID.....	27
Gambar 2. 28 Diagram Blok PID .....	27
Gambar 2. 29 Osilasi PID.....	28
Gambar 2. 30 Fungsi Alih kontroller .....	28
Gambar 2. 31 Kurva Respon .....	29
Gambar 2. 32 Persamaan Kurva Respon.....	29
Gambar 3. 1 General Arrangement Kapal .....	33
Gambar 3. 2 Model Kapal .....	33
Gambar 3. 3 Arduino.....	34
Gambar 3. 4 Sensor Ultrasonik .....	34
Gambar 3. 5 Gyro Accelerometer .....	35
Gambar 3. 6 Motor DC.....	35
Gambar 3. 7Kabel .....	35
Gambar 3. 8 ESC.....	36
Gambar 3. 9 Servo.....	36
Gambar 3. 10 Baterai.....	36
Gambar 3. 11 Pengujian 1 .....	37
Gambar 3. 12 Pengujian 2 .....	37

Gambar 4. 1 Blok Diagram Sistem.....	41
Gambar 4. 2 Blok Diagram Sistem 2.....	42
Gambar 4. 3 General Arrangement .....	43
Gambar 4. 4 Model Kapal .....	43
Gambar 4. 5 Spesifikasi Motor.....	47
Gambar 4. 6 Wiring Diagram .....	49
Gambar 4. 7 Skematik Elektronik .....	50
Gambar 4. 8 Thruster Penggerak.....	51
Gambar 4. 9 Sensor Ultrasonik.....	51
Gambar 4. 10 Servo.....	51
Gambar 4. 11 Rangkaian Elektronik .....	52
Gambar 4. 12 Hasil Akhir Model .....	52
Gambar 4. 13 Kolam Uji .....	74
Gambar 4. 14 Posisi Pengujian.....	74
Gambar 4. 15 Pengujian 1 .....	75
Gambar 4. 16 Pengujian 2 .....	75
Gambar 4. 17 Kurva Arduino .....	77
Gambar 4. 18 Kurva Arduino 20 cm .....	79
Gambar 4. 19 Kurva Arduino 20 cm .....	80
Gambar 4. 20 Kurva Arduino 40 cm .....	80
Gambar 4. 21 Kurva Arduino 40 cm .....	80
Gambar 4. 22 Kurva Arduino 60 cm .....	81
Gambar 4. 23 Kurva Arduino 60 cm .....	81
Gambar 4. 24 Kurva Arduino 80 cm .....	81
Gambar 4. 25 Kurva Arduino 80 cm .....	82
Gambar 4. 26 Kurva Arduino 100 cm .....	82
Gambar 4. 27 Kurva Arduino 100 cm .....	82

## DAFTAR TABEL

Tabel 2. 1 Keunggulan dan kekurangan Dynamic Positioning .....	4
Tabel 2. 2 Notasi ABS.....	14
Tabel 2. 3 Dimensi Kapal.....	15
Tabel 2. 4 Perbedaan Sinyal Analog dan Digital .....	20
Tabel 2. 5 Karakteristik PID.....	25
Tabel 2. 6 Tabel Permsamaan Siegler Nichols.....	28
Tabel 2. 7 Penentuan Koefisien.....	29
Tabel 3. 1. Dimensi Utama Kapal .....	32
Tabel 4. 1 Komponen DP System .....	41
Tabel 4. 2 Dimensi Utama Kapal .....	42
Tabel 4. 3 Pin Arduino .....	44
Tabel 4. 4 Spesifikasi SRF .....	45
Tabel 4. 5 Pin SRF .....	45
Tabel 4. 6 Spesifikasi Gyro- Accelerometer .....	46
Tabel 4. 7 Pin Gryro .....	46
Tabel 4. 8 Spesifikasi ESC .....	48
Tabel 4. 9 Spesifikasi Servo .....	48
Tabel 4. 10 Spesifikasi Baterai.....	49
Tabel 4. 11 Hasil Trial And Error .....	78
Tabel 4. 12 Tabel PWM .....	78
Tabel 4. 13 Hasil Pengujian 1 .....	79
Tabel 4. 14 Hasil Pengujian 2 .....	83

“ Halaman ini sengaja dikosongkan”

# BAB I

## PENDAHULUAN

### 1. 1. Latar Belakang

Dynamic positioning (*DP System*) merupakan suatu sistem pengendalian yang dikendalikan oleh komputer secara otomatis untuk mempertahankan posisi dan pos (*Heading*) suatu kapal/bangunan laut dengan menggunakan baling-baling dan pendorong (*Thruster*). DP system pada umumnya terdiri dari sistem referensi berupa referensi posisi dan arah menggunakan sensor, sistem kontrol dan komputer untuk mengolah data, elemen pendorong berupa thruster (*Thruster System*), serta sistem pembangkit daya (*Power System*). Teknologi Dynamic Positioning semakin berkembang dan mulai banyak digunakan, tidak hanya pada industri minyak dan gas lepas pantai, namun juga pada berbagai jenis kapal dan kegiatan lain. Dynamic Positioning digunakan untuk banyak aplikasi diantaranya, kapal platform supply, kapal crane, kapal pesiar, kapal dukungan diving, pengerukan, drill - kapal, FPSO, Flotels, Landing platform Docks, Penelitian atau Survey Kapal, batu-pembuangan dll. Pengaplikasian Dynamic Positioning di Indonesia berkembang seiring dengan perkembangan industri minyak dan gas di Indonesia.

Platform *Supply vessel* merupakan jenis kapal yang digunakan untuk menunjang kegiatan eksplorasi dan produksi lepas pantai. Kapal ini beroperasi di perairan sekitar daerah *offshore / platform* dan juga area labuh. Kapal ini digunakan mengangkut berbagai jenis kebutuhan logistik dari dan menuju oil /gas *platform*. Kapal harus mampu membawa sejumlah besar muatan pada kecepatan tertentu, jarak dan mampu beroperasi dengan aman. Dynamic Positioning System digunakan pada *Supply Vessel* untuk mempertahankan posisi kapal selama beroperasi. Selama kapal beroperasi di sekitar rig/platform, kemampuan kapal untuk mempertahankan posisi diam, berguna untuk mencegah kapal berbenturan dengan badan platform selama proses *Loading Unloading* berlangsung ataupun bergerak menjauhi platform. Gaya yang diterima kapal berasal dari gaya luar dari lingkungan yaitu arus air, ombak maupun angin.

Pada Dynamic Positioning berbasis PID, Sistem kontrol dan komputer akan mengendalikan respon keluaran dari sistem penggerak. Input Sistem kontrol berupa sinyal dari Sistem Referensi yang kemudian dimasukkan nilai set point (masukan yang diinginkan). Proses yang terjadi pada Sistem kontrol adalah membandingkan kedua sinyal input, perbedaan nilai akan menjadi nilai error. Sistem kontrol akan mengolah data berbasis PID dan mengirimkan perintah untuk mengeliminir nilai error. Perintah dikirim dan dijalankan oleh sistem penggerak. Gaya yang diberikan dari sistem penggerak akan mempertahankan posisi kapal akibat daya yang diberikan dari lingkungan.

Penelitian kali ini dilakukan untuk merancang dan melakukan pengujian dari Dynamic Positioning system pada skala universitas untuk menyelesaikan permasalahan yang muncul pada supply Vessel dalam rangka pemenuhan fungsi kerja. Pada penelitian kali ini akan dilakukan perancangan dan pemograman sistem kendali Dynamic positioning menggunakan sensor ultrasonik dengan Kontrol Arduino berbasis PID. Dari hasil perancangan akan dilakukan simulasi pada model Supply Vessel dengan menggunakan arus air sebagai gaya dari lingkungan.

### 1. 2. Rumusan Masalah

Pokok permasalahan yang akan dibahas adalah :

- Bagaimana mendesain dan merancang system sensorik Dynamic positioning pada model supply vessel dengan Kontrol Arduino ?
- Bagaimana merancang program kendali Dynamic Positioning dengan Kontrol Arduino berbasis PID ?

### 1. 3. Batasan Masalah

Agar lingkup penelitian ini menjadi terfokus maka batasan masalah yang diterapkan adalah :

- Sensor referensi yang digunakan yaitu sensor ultrasonic SRF HC-04 dan Gyro-Accelerometer MPU 5060.
- Hasil simulasi dibatasi pada keberhasilan program untuk memenuhi kebutuhan minimal Dynamic Positioning akibat gaya dari arus air pada supply vessel. Variabel gaya lain dari lingkungan seperti angin dan ombak diabaikan.
- Pada pengujian kemampuan mempertahankan posisi yang diuji pada model berupa 2 derajat kebebasan yaitu gerakan sway dan yaw.
- Perancangan dan pembuatan model dibuat pada skala simulasi dan dapat mengilimir variable nyata lain yang ada di keadaan sebenarnya seperti tahanan, gaya hidrodinamik dan lain lain.

### 1. 4. Tujuan

Dari rumusan masalah di atas maka tujuan penulisan skripsi ini adalah :

- Membuat alat rancang bangun dynamic positioning system model model supply vessel menggunakan sensor ultrasonik dengan kontrol Arduino.
- Menganalisa respon kontrol PID dan hasil pengujian pada system kendali Dynamic positioning pada model supply vessel .
- Mengetahui Nilai input tuning PID berdasar metode Ziegler-Nichols dan kecepatan putaran motor untuk mengatasi gaya lingkungan akibat arus air pada berbagai keadaan uji.

## BAB II TINJAUAN PUSTAKA

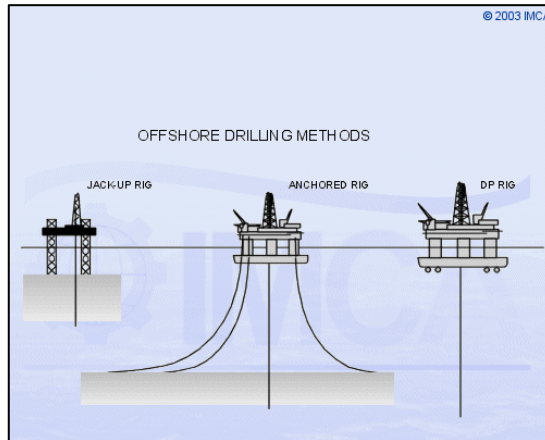
### 2. 1. Dynamic Positioning

*Dynamic positioning* (DP) merupakan suatu sistem pengendalian yang dikendalikan oleh komputer secara otomatis untuk mempertahankan posisi dan pos (*Heading*) suatu kapal/bangunan laut dengan menggunakan baling-baling dan pendorong (*Thruster*). Sensor referensi posisi dikombinasikan dengan sensor referensi pos (*Heading*) berupa girokompas, sensor angin dan sensor gerak. Sensor memberikan informasi ke komputer yang berkaitan dengan posisi kapal dan besar serta arah gaya akibat lingkungan yang mempengaruhi posisinya. (The Nautical Institute,2011).

Program komputer berisi model matematika dari kapal mencakup informasi yang berkaitan dengan arah angin, daya tarik arus dan lokasi pendorong. Hal ini dikombinasikan dengan informasi sensor, yang memungkinkan komputer untuk menghitung sudut kemudi yang diperlukan dan keluaran pendorong (*Thruster*). *Dynamic positioning* bisa bersifat absolut yaitu dalam posisi terkunci tetap ke sebuah titik acuan, ataupun bergerak dengan respon relatif ke objek lain seperti kapal atau kendaraan bawah air lain. (The Nautical Institute,2011).

Dynamic Positioning diterapkan pada operasi di laut dimana system tambat (*Mooring System*) tidak memungkinkan untuk digunakan karena kedalaman laut, adanya objek tertentu di dasar laut atau alasan teknis lain. Dynamic Positioning banyak digunakan dalam industri minyak dan gas lepas pantai. Dynamic Positioning diaplikasikan pada *platform supply vessel, Mobile Offshore Drilling Unit, Oceanographic research vessels, Cable layer ships* dan kapal pesiar. (The Nautical Institute,2011).

Dynamic Positioning mulai dikenal di awal tahun 1960 seiring dengan meningkatnya kegiatan industri dan eksplorasi minyak dan gas untuk pengeboran lepas pantai. Dengan meningkatnya kegiatan industri dan eksplorasi, kegiatan pengeboran minyak lepas pantai, mulai bergerak ke perairan yang lebih dalam dengan kondisi lingkungan yang lebih sulit. Dengan keadaan seperti ini, bangunan lepas pantai jenis *Jack up Barges* sudah tidak dapat lagi digunakan dan teknik *anchoring* menjadi kurang ekonomis. Sejak saat itu metode Dynamic Positioning mulai dikembangkan, dan kemudian menjadi tuntutan bagi setiap industri pengeboran minyak dan gas. Teknologi Dynamic Positioning semakin berkembang dan mulai banyak digunakan, tidak hanya pada industri minyak dan gas lepas pantai, namun juga pada berbagai jenis kapal dan kegiatan lain. Sampai saat ini sudah terdapat lebih dari 1000 bangunan laut yang menggunakan teknologi Dynamic Positioning. (International Marine Contractors Association,2003)



Gambar 2. 1. Metode Pengeboran Bangunan Lepas Pantai  
 Sumber : International Marine Contractors Association, 2003

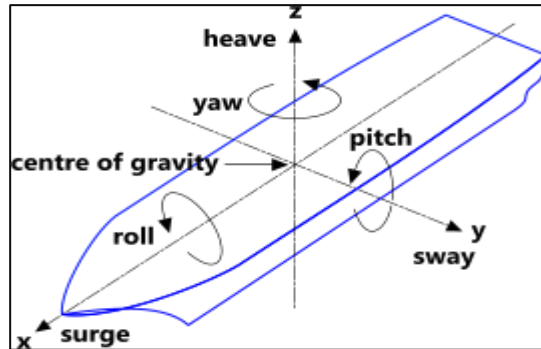
Seperti yang disebutkan sebelumnya, metode lain yang digunakan untuk menjaga posisi tetap pada bangunan laut antara lain *Spread and fixed moorings* dan Jack-ups. Metode *Spread and fixed moorings* juga dapat dilengkapi dengan sistem kontrol DP untuk membantu pengaturan posisi untuk mengurangi ketengan antar tali tambat.

Tabel 2. 1 Keunggulan dan kekurangan Dynamic Positioning  
 Sumber : International Marine Contractors Association, 2003

Jack-up barge	Anchoring	Dynamic positioning
Kelebihan : Sistem Lebih sederhana Tidak ada resiko perpindahan posisi karena kegagalan sistem	Kelebihan : Sistem Lebih sederhana Tidak ada resiko perpindahan posisi karena kegagalan sistem	Kelebihan : Manuvering yang lebih baik. Tidak memerlukan penanganan dari jangkar atau kapal tunda Pembangunan bangunan lepas pantai tidak dibatasi oleh kedalaman dan konstruksi dasar laut
Kekurangan : Tidak dapat bermanuver. Terbatas pada kedalaman laut 175 meter	Kekurangan : Manuver terbatas Membutuhkan jangkar Kurang cocok pada perairan dalam Dibatasi oleh struktur lain di bawah air. (Pipa bawah laut)	Kekurangan Sistem lebih kompleks Biaya Instalasi awal, operasional dan pemeliharaan yang lebih tinggi Kemungkinan terjadinya kegagalan sistem lebih besar

Seiring dengan perkembangan teknologi, biaya instalasi dan operasional DP system menjadi semakin murah dan dapat menjangkau perairan yang lebih dalam. Teknologi DP semakin menjadi pilihan utama pada industry minyak dan gas dunia.





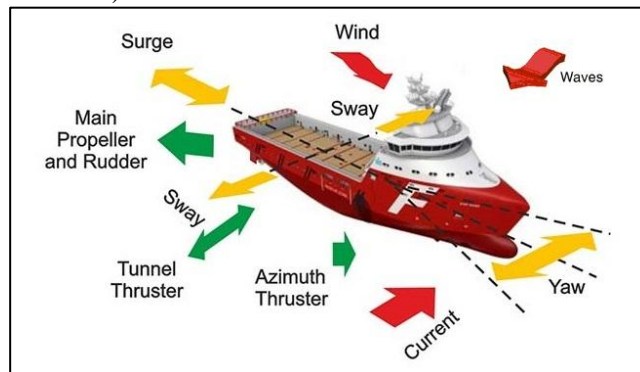
Gambar 2. 2. Enam Derajat Kebebasan Kapal  
Sumber : calqlata.com

Kapal memiliki 6 derajat kebebasan pada pergerakannya. 3 gerakan mencakup gerakan translasi yaitu :

- Surge ( Forward/stern)
- Sway ( Starboard/port)
- Heave (Up/ Down)

Tiga gerakan lain mencakup gerakan rotasi yaitu :

- Roll ( pada sumbu Y)
- Pitch ( pada sumbu X)
- Yaw (pada sumbu Z)



Gambar 2. 3. Derajat Kebebasan Dynamic Positioning Supply Vessel  
Sumber : www.offshoreengineering.com

Dynamic Positioning dapat mengontrol gerakan kapal pada bidang horizontal. Gerakan itu mencakup Surge, Sway, dan Yaw.

Kapal yang menggunakan DP system memerlukan :

- Sistem Referensi untuk bisa mempertahankan posisi dan arah, posisi dan arah kapal harus diketahui.
- Komputer dan sistem control untuk menghitung pergerakan yang diperlukan dalam mempertahankan posisi dan mengeliminir nilai error.
- Elemen Pendorong untuk memberikan gaya yang dibutuhkan berdasarkan system control serta sistem daya.

### 2.1.1 Sistem Referensi

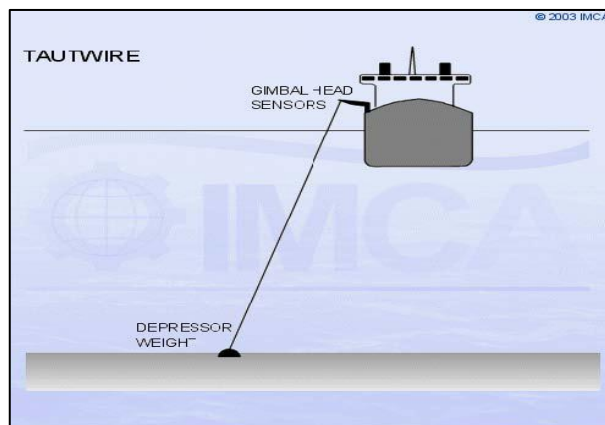
Seperti yang dijelaskan sebelumnya, sistem Referensi diperlukan untuk bisa mempertahankan posisi dan arah, posisi dan arah kapal harus diketahui dan dikolaborasikan dengan gaya lain dari faktor lingkungan dan kapal sendiri. Menurut fungsinya system referensi dikelompokkan menjadi :

#### 1) Sistem Referensi Posisi

Ada beberapa metode untuk menentukan posisi kapal. Peralatan yang sering digunakan pada navigas kapal tidak cukup akurat dipakai pada dynamic positioning. Untuk itu perlatan system referensi untuk Dynamic Positioning telah dikembangkan selama decade terakhir. DP system bisa menggunakan lebih dari satu sistem sensor referensi posisi / Position Reference System (PRS) yang mana digunakan sesuai kemampuan alat masing masing. Untuk DP yang mewajibkan redudansi (DP kelas 1 dan 3), dibutuhkan minimal 3 referensi posisi dengan jenis yang berbeda. Hal ini diharuskan agar setidaknya dua sensor bisa mengidentifikasi adanya kerusakan/kegagalan dari satu sensor lain. Beberapa produsen peralatan system Dynamic Positioning diantaranya Marine Technologies LLC, Kongsberg Maritime, Navis Engineering Oy, GE, DCNS, Wartsila (ex L-3), MT-div.Chouest, Rolls-Royce plc, dan Praxis Automation Technology. Berikut beberapa system referensi posisi yang sering digunakan:

#### a) Light Taut Wire

Metode ini digunakan saat suatu bangunanan laut ingin menghabiskan waktu yang lama pada keadaan statis. Peralatan yang digunakan berupa derek yang ditempatkan pada sisi dek, pemberat (*clumpweight*) yang dihubungkan dengan kawat kemudian diturunkan ke dasar laut. Terdapat sensor pada kepala gimbal yang digunakan untuk mengukur sudut kawat. Setelah beban diturunkan, derek berada pada posisi tegangan konstan. Kemudian derek beroperasi untuk mempertahankan tegangan kawat dan mendeteksi gerakan kapal. Posisi relative diukur berdasarkan panjang kawat yang diulur dan sudut terhadap kepala gimbal. Sistem ini memiliki keterbatasan sudut kawat dan kurang cocok digunakan pada perairan yang dalam. (International Marine Contractors Association,2003).

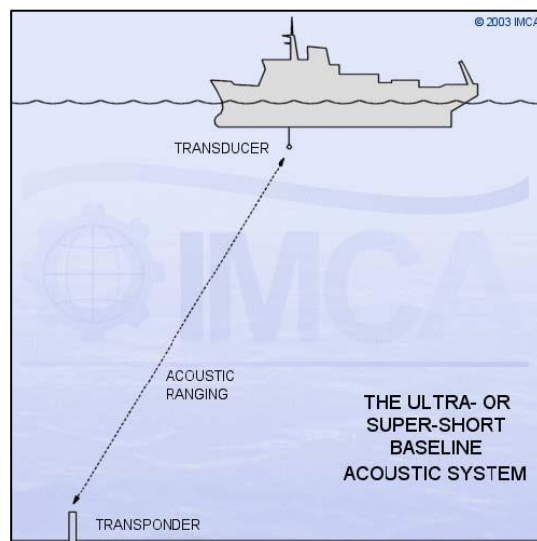


Gambar 2. 4 Light Taut Wire

Sumber : International Marine Contractors Association,2003

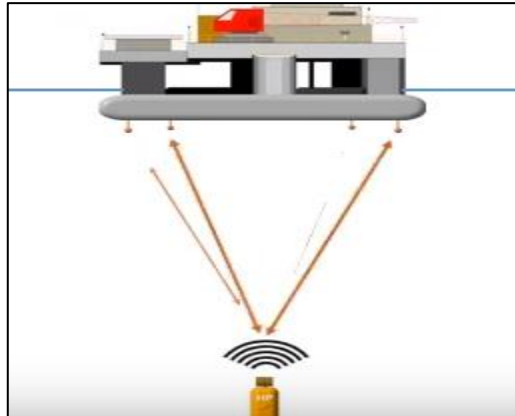
### b) Sensor Akustik Bawah Air

Sistem ini menggunakan transduser yang ditempatkan bagian lunas pada lambung kapal dan transponder yang ditempatkan pada dasar laut. Transduser mengirim sinyal akustik ke transponder, dan memicu sinyal umpan balik. Jarak didapatkan berdasarkan kecepatan rambat gelombang suara pada media air, sehingga posisi kapal terhadap transponder diketahui. Sensor Hydroacoustic Position Reference (HPR) juga dapat digunakan untuk melacak keberadaan objek bawah air, menandakan posisi benda bawah air dan mengontrol peralatan bawah air. Kekurangan system ini adalah rentan adanya gangguan gelombang dari thruster dan peralatan akustik lain. Penggunaan juga terbatas pada perairan yang dangkal karena pembiasan gelombang pada media rambat saat merambat secara horizontal. (The Basic Principles and Use of Hydroacoustic Position Reference Systems in the Offshore Environment, 1999)



Gambar 2. 5. Ultra / Super Short baseline ( USBL /SSBL)  
Sumber : International Marine Contractors Association,2003

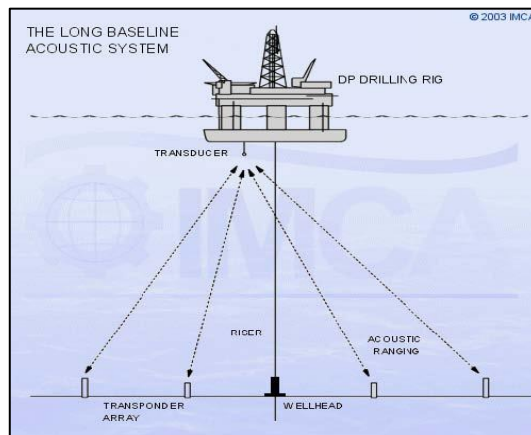
- Ultra / Super Short baseline ( USBL /SSBL)  
Menggunakan satu transponder dan satu transduser. Frekuensi yang dihasilkan berkisar 20-30 kHz. Hasil pembacaan sensor berupa sudut horizontal dan vertical serta jarak kapal berdasarkan respon waktu dari transduser. Pembacaan sudut dari transponder dijadikan koreksi untuk *pitch* dan *roll* dari kapal yang ditentukan oleh Motion Reference Unit. Kemampuan terbatas pada kondisi akustik bawah laut dengan adanya gangguan gelombang dari peralatan lain ataupun turbulensi dari ombak dan arus bawah air. (International Marine Contractors Association,2003).



Gambar 2. 6. Short baseline (SBL)

Sumber : [www.dynamicpositioning.guru](http://www.dynamicpositioning.guru)

- Short baseline (SBL)  
Menggunakan serangkaian setidaknya 3 transducer atau lebih yang dipasang pada bangunan laut. Posisi penempatan dari transducer merupakan referensi untuk posisi tetap kapal. Penentuan posisi dari transducer dilakukan pada kontrol dimensi kapal. (The Basic Principles and Use of Hydroacoustic Position Reference Systems in the Offshore Environment, 1999)



Gambar 2. 7. Long Baseline (LBL)

Sumber : International Marine Contractors Association, 2003

- Long Baseline (LBL)  
Menggunakan setidaknya 3 atau lebih transponder. Posisi kapal secara teori berada di antara pertemuan radius imajiner antar transponder. Sudut dari pengukuran tidak dibutuhkan karena jarak antar transponder sudah dikalibrasi sebelumnya, sehingga akurasi pada kedalaman air yang dalam (>1000 m) lebih baik dari sistem USBL. (International Marine Contractors Association, 2003).

### c) Sensor Gelombang

Sensor bekerja dengan memanfaatkan sifat pantulan gelombang untuk mengukur jarak dan menentukan posisi kapal terhadap rig. Beberapa jenis sensor gelombang antara lain :

- **SONAR ( Sound Navigation and Ranging)**  
Sistem ini memiliki prinsip kerja yang sama dengan sensor gelombang lainnya yaitu memanfaatkan prinsip pantulan gelombang. Karakteristik dasar yang membedakan sistem SONAR dengan prinsip sensor gelombang lain, terletak pada jenis gelombang yang ditransmisikan. Sistem SONAR mentransmisikan gelombang suara yang merupakan jenis dari gelombang mekanik. Frekuensi akustik yang digunakan pada SONAR bervariasi dari frekuensi rendah (Infrasonic) sampai frekuensi tinggi ( Ultrasonik). Transmisi gelombang suara juga merupakan prinsip yang digunakan pada sistem sensor akustik bawah air. (*Hydroacoustic*). Gelombang suara yang merupakan gelombang mekanik membutuhkan media perambatan gelombang. Pada perkembangannya sensor sonar lebih banyak digunakan pada aplikasi bawah air berupa sensor akustik bawah air. Hal ini dikarenakan, pada umumnya gelombang suara lebih mudah untuk merambat pada media rambat air dibandingkan udara. Walaupun kecepatan rambat gelombang suara lebih lambat pada media air dibandingkan udara, namun gangguan (Noise) pada penggunaan media air lebih mudah untuk dieliminir.



Gambar 2. 8. RadaScan Pada Kapal  
Sumber : [www.guidance.eu.com](http://www.guidance.eu.com)

- **RADAR ( Radio Detection and Ranging)**  
RADAR mentransmisikan jenis gelombang radio yang merupakan jenis gelombang elektromagnetik. Gelombang elektromagnetik, merupakan jenis gelombang yang tidak membutuhkan media rambat. Kecepatan gelombang radio dan luas area yang dapat dijangkau pada RADAR lebih baik ketimbang gelombang suara pada SONAR. Pada aplikasi Dynamic Positioning, sistem ini bekerja dengan mengirimkan gelombang mikro dari unit pemancar/interogator

yang berada pada kapal. Kemudian gelombang tersebut ditangkap oleh unit penangkap / responder pada rig dan mengirimkan gelombang pantulan kembali ke unit di kapal. Respon gelombang pantulan digunakan untuk menentukan posisi kapal dan sudut terhadap responder pada rig. Contoh sistem sensor yang menggunakan prinsip ini adalah Artemis, RADIUS, dan RadaScan. RADIUS dan RadaScan tidak mempunyai bagian yang bergerak seperti Artemis. Ukuran transponder lebih kecil dibanding Artemis dengan jangkauan antara 500-1000 meter. (A Review Of The Artemis Mark V, 2004)



*Gambar 2. 9 FanBeam Laser*  
Sumber : renishaw.com

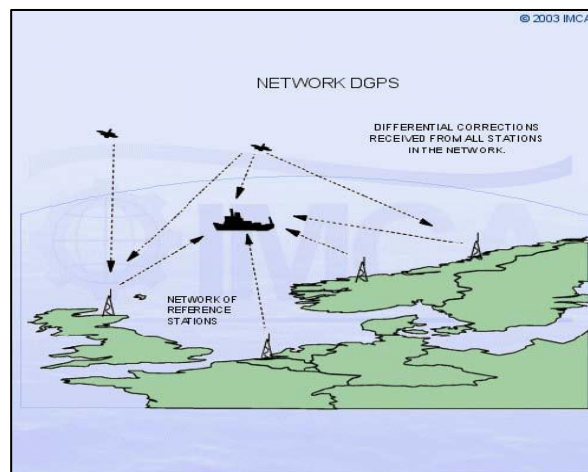
- LIDAR

LIDAR mentransmiskan jenis gelombang cahaya berupa laser. Prinsip kerja LIDAR sama dengan sensor gelombang lainnya, yaitu memanfaatkan prinsip pantulan gelombang. Pada kapal unit transmisi gelombang dipasang, dan bekerja dengan menghasilkan laser yang ditunjukkan kepada unit pemantul. Unit pemantul cahaya laser dipasang pada rig. Unit pemantul gelombang cahaya pada rig bisa berbentuk lempengan, silinder ataupun prisma. Kekurangan dari sistem ini adalah resiko gangguan gelombang umpan balik akibat objek pemantul cahaya lain di sekitar struktur rig. Contoh sensor laser adalah Fanbeam dan Cyscan. Pada CyScan, sensor berputar setiap satu detik. (A Review of Marine Laser Positioning Systems, 2003).

#### d) Differential Global Positioning System (DGPS)

Referensi posisi hasil pembacaan GPS memiliki tingkat akurasi yang rendah. Untuk meningkatkan tingkat akurasi GPS, koreksi diferensial diperlukan pada GPS data. Pada system ini, data posisi hasil pembacaan GPS pada kapal di bandingkan dengan data stasiun acuan ( Differential Station) di darat . Hasil pembacaan dikirim ke penerima DGPS dengan frekuensi radio. Kelebihan dari DGPS adalah selalu tersedia dimanapun. Kekurangan DGPS adalah degradasi sinyal oleh gangguan di atmosfer, gangguan dari crane dan struktur bangunan tinggi lain ( Guidelines on the Use of DGPS, 1997)

Sebagian besar layanan DGPS menerima sinyal deferensial yang diperoleh dari berbagai stasiun referensi yang terpisah. Dengan ini jaringan akan lebih stabil dan meningkatkan akurasi. Sistem jaringan dapat dimonitor pada Hub atau stasiun kontrol. Sebagian DP



Gambar 2. 10. Network DGPS configuration

Sumber : International Marine Contractors Association,2003

#### 2) Sistem Referensi Arah

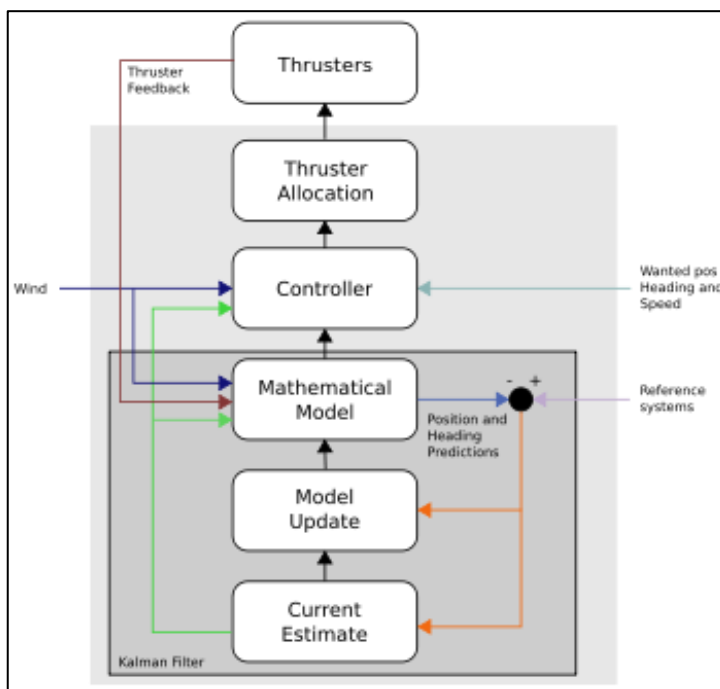
Sistem referensi arah / Heading references diperlukan untuk menentukan posisi arah kapal. Sensor yang biasa digunakan pada DP system adalah Gyro Compasses. Jika kapal membutuhkan redundansi maka lebih dari satu gyro compasses dipasang di kapal. Grycompass adalah tipe non magnetic kompas. Komponen terpenting pada gyrocompass adalah Gyroscope.

#### 3) Sistem Referensi Lingkungan

Selain sensor referensi arah dan posisi, sensor tambahan lain juga diperlukan akibat adanya gaya dari lingkungan kapal. Gaya tersebut bisa didapatkan dari angin dan ombak. Gaya dari ombak dapat mempengaruhi kemiringan kapal dan tinggi dari sarat kapal. Beberapa contoh sensor tambahan antara lain sensor Motion reference units, vertical reference units or vertical reference sensors, VRUs or MRUs or VRSs yang mengatur gerakan roll, pitch dan heave. Sensor angin dimasukkan ke dalam sistem DP feedforward, sehingga sistem dapat mengantisipasi hembusan angin sebelum kapal berubah posisi. Sensor draft, karena perubahan draft mempengaruhi efek angin dan arus pada lambung.

### 2.1.2 Sistem Kontrol

Setiap kapal menerima gaya dari angin, gelombang dan gerakan pasang surut serta gaya yang dihasilkan dari sistem propulsi dan elemen eksternal lainnya. Respon terhadap gaya ini adalah gerakan kapal, yang mengakibatkan perubahan posisi dan arah. Perubahan ini diukur dengan sensor sistem referensi posisi dan Gyro Kompas. Sistem kontrol DP menentukan selisih antara nilai posisi dan arah yang terukur/real dengan nilai yang dibutuhkan/ditentukan (SetPoint), dan menghitung gaya yang harus dihasilkan oleh pendorong untuk mengurangi selisih tersebut mencapai nol. Selain itu sistem kontrol DP juga harus menghitung kekuatan angin yang bekerja di atas kapal, dan gaya dorong yang diperlukan untuk mengatasi gaya tersebut berdasarkan model kapal yang dibuat di komputer. Kapal akan berusaha terus mempertahankan posisi secara otomatis, meskipun penjagaan posisi akan memburuk dengan meningkatnya waktu yang dihitung sejak data posisi terakhir kapal diterima. Dalam istilah praktis, ini berarti bahwa DPO tidak perlu segera memilih "manual" kontrol atas hilangnya semua referensi posisi.



Gambar 2. 11. Diagram Blok Sistem Kontrol

Sumber : International Marine Contractors Association,2003

Sistem kontrol DP, awalnya terdiri dari kontroler PID analog sederhana yang tidak beradaptasi dengan kondisi laut yang sebenarnya. Perkembangan metode Kalman dan transmisi data digital yang semakin cepat telah memberikan beberapa perbaikan signifikan pada akurasi sistem.

Kontroler PID yang digunakan dulu masih digunakan dalam sistem DP sampai saat ini. Sistem pengendali modern menggunakan Model matematis kapal yang didasarkan pada gaya hidrodinamik dan aerodinamis serta karakteristik kapal seperti massa dan drag.



Tentu saja, model ini tidak sepenuhnya benar. Posisi kapal dan arah dimasukkan ke dalam sistem komputer dan dibandingkan dengan data prediksi yang dibuat oleh model. Perbedaan ini digunakan untuk memperbarui model dengan menggunakan teknik perhitungan Kalman. Akurasi dan ketepatan pada tiap sensor yang berbeda tidak sama. DGPS memiliki akurasi dan presisi yang tinggi, dan metode USBL memiliki presisi yang jauh lebih rendah.

### **2.1.3 Sistem Daya**

Bagian penting dari operasi kapal dengan DP system adalah pembangkit daya, penyaluran dan sistem distribusi daya. Daya digunakan untuk menggerakkan Thruster dan sistem pendukung lainnya, serta semua elemen dp control dan sistem referensi. Thruster pada kapal pada umumnya bagian yang paling banyak memerlukan supply daya. Sistem kontrol akan lebih banyak membutuhkan banyak suplai daya pada saat kondisi lingkungan yang sulit. Pembangkit daya harus bersifat fleksibel dalam menyediakan sumber listrik di kapal tanpa menghabiskan konsumsi bahan bakar berlebih. Pada umumnya kapal DP menggunakan pembangkit listrik tenaga diesel dengan sistem pendorong dan alat alat elektrik lain disuplai oleh alternator diesel yaitu generator diesel. Beberapa kapal DP terdiri dari bagian penggerak tenaga diesel dan tenaga listrik. Sebuah kapal mungkin memiliki 2 propeller sebagai pendorong utama oleh mesin diesel dan bow/stern thruster digerakkan secara elektrik dengan tenaga dari alternator yang digerakkan oleh mesin diesel utama atau dari Diesel Generator yang terpisah.

Pada DP kelas 2 atau 3, Sistem kontrol DP terlindung dari kegagalan daya listrik dengan disertakannya Uninterruptible Power Supply (UPS). Sistem ini menyediakan daya yang stabil yang tidak terpengaruh oleh interupsi jangka pendek atau fluktuasi daya AC kapal. Daya memasok listrik untuk komputer, konsol kontrol, monitor, alarm dan sistem referensi. Jika terjadi gangguan pada pasokan listrik AC utama kapal, baterai cadangan akan memasok daya ke semua sistem ini selama minimal 30 menit. (International Marine Contractors Association,2003)

### **2.1.4 Sistem Propulsi**

Kemampuan mempertahankan posisi kapal dilakukan oleh thruster/penggerak. Secara umum jenis thruster yang digunakan yaitu, propeller utama, tunnel thruster dan azimuth thruster. Propoller utama dapat berupa single atau double screw layaknya kapal konvensional lain. Pada kapal, jika penggerak utama merupakan bagian dari sistem Dynamic Positioning maka, jenis propeller yang digunakan adalah controllable pitch propeller disertai rudder yang dipasang sistem kontrol dan feedback.

Selain itu DP sistem juga harus memiliki thruster untuk mengontrol posisi. Pada kapal jenis monohull, biasanya dilengkapi dengan 6 thruster, tiga pada bagian deoan dan tiga lagi di belakang. Thruster bagian depan berbentuk tunnel thruster. Thruster bagian belakang biasanya dapat dikendalikan secara terpisah semisal azimuth propeller. Azimuth propeller harus dapat digerakkan ke segala arah. Pada beberapa jenis kapal, azimuth propeller juga digunakan sebagai penggerak utama kapal. (International Marine Contractors Association,2003)

## 2. 2. Classification Notation

Berdasarkan ABS tentang *dynamic positioning system* pada vessel, sistem Dynamic Positioning memiliki beberapa pengelompokan notasi sesuai dengan klasifikasi antara lain :

- **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.
- **DPS-1**  
Untuk kapal, yang dilengkapi dengan sistem penentuan posisi dinamis yang mampu secara otomatis mempertahankan posisi dan arah kapal di bawah kondisi lingkungan yang maksimum yang ditentukan memiliki sistem kontrol posisi manual.
- **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 komparteme
- **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 kerusakan atau banjir.

Berikut merupakan table perbandingan sistem Dynamic Positioning pada masing masing notasi :

*Table 2. 2 Notasi ABS*

Sumber : marine insight

<b>TABLE 1</b>						
<b>Summary of DP System Requirements for ABS DPS Notations<sup>(4)</sup> (1 November 2013)</b>						
Subsystem or Component	Equipment	Minimum Requirements for each Classification Notation				
		DPS-0 <sup>(1)</sup>	DPS-1	DPS-2	DPS-3 <sup>(5)</sup>	
Power System	Generators and Prime Movers	Non-redundant	Non-redundant	Redundant	Redundant, in separate compartments <sup>(5)</sup>	
	Main Switchboard	1	1	1 with bus-tie	2 with bus-ties, in separate compartments	
	Bus-tie Breaker	0	0	1	2	
	Distribution System	Non-redundant	Non-redundant	Redundant	Redundant, in separate compartments	
	Power Management <sup>(2)</sup>	No	No	Yes	Yes	
Thrusters	Arrangement of Thrusters	Non-redundant	Non-redundant	Redundant	Redundant, in separate compartments	
Control System	DP Control: Number of Control Computers	0	1	2	2 + 1 in backup control station	
	Manual Position Control: Joystick with Auto Heading	Yes	Yes	Yes	Yes	
	Manual Thruster Control	Yes	Yes	Yes	Yes	
	Position Reference Systems	1	2	3	2 + 1 in backup control station	
	Sensors:	Wind	1	2	3	2 + 1 in backup control station
		MRU <sup>(3)</sup>	0	1	3	2 + 1 in backup control station
		Gyro	1	2	3	2 + 1 in backup control station
	UPS	0	1	2	2 + 1 in separate compartment	
Backup Control Station for Backup Unit		N/A	N/A	N/A	Yes	
Consequence Analyzer		No	No	Yes	Yes	

### 2. 3. Supply Vessel



*Gambar 2. 12 Supply Vessel*  
Sumber : Edison Chouest Offshore

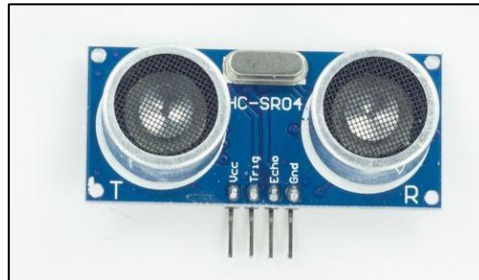
Platform Supply vessel merupakan jenis kapal yang digunakan untuk menunjang kegiatan eksplorasi dan produksi lepas pantai. Kapal ini beroperasi di perairan sekitar daerah offshore / platform dan juga area labuh. Kapal ini digunakan mengangkut berbagai jenis kebutuhan logistik dari dan menuju oil /gas platform. Kapal harus mampu membawa sejumlah besar muatan pada kecepatan tertentu, jarak dan mampu beroperasi dengan aman. Berikut data kapal supply vessel yang digunakan pada tugas akhir.

*Tabel 2. 3 Dimensi Kapal*

Parameter	Kapal Asli	Kapal Model
Length Overall	26.70 m	1.02 m
LBP	22.95 m	
Height		
Draught	1.4 m	0.5 m
Beam Hull ( Hull)	6.00 m	0.9 m
Beam ( Overall)	6.50 m	0.3 m
Fuel Capacity	17.0 m3	
Fresh Water Capacity	5.0 m3	

## 2. 4. Sensor Ultrasonik

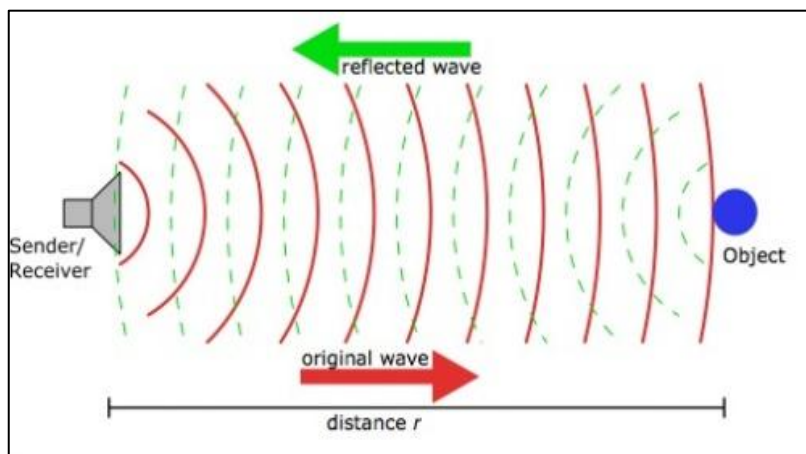
Sensor Ultrasonik, merupakan sensor gelombang yang memanfaatkan prinsip pantulan gelombang. Sensor bekerja berdasarkan prinsip kerja sistem sensor SONAR, RADAR dan LIDAR. Sensor ultrasonik memanfaatkan gelombang suara. Sensor ultrasonik yang umum digunakan pada bidang elektronik adalah sensor jenis SRF ( Self Resonant Freuency). Salah satu tipe dari sensor SRF adalah sensor SRF HC 04.



Gambar 2. 13 SRF HC-04

Sumber : Amazon.in

Sensor dapat mendeteksi objek di sekitar sensor dengan kisaran rentang jarak 2-400 cm. Sensor bekerja dengan memberikan sinyal keluaran melalui transmitter (*Trigger Pulse*) dengan panjang gelombang 10 microsecond yang memicu 8 gelombang ultrasonic sebesar 40 KHz. SRF bekerja dengan Prinsip Time Of Flight / Time Delay (TOF) Pantulan sinyal balik akan diterima oleh receiver (Echo Pulses) dengan rentang panjang gelombang 150 uS sampai 25 mS.



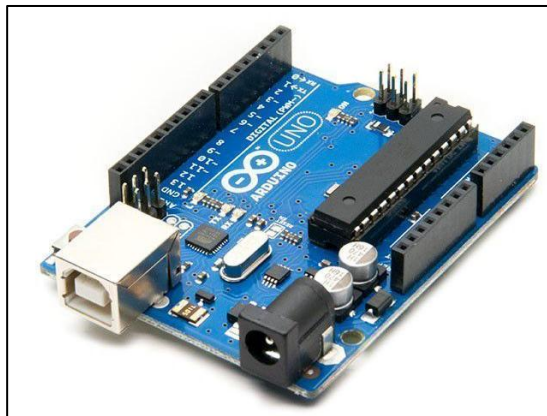
Gambar 2. 14 Prinsip Kerja SRF

Sumber : wiki.eprolabs.com

Gelombang ultrasonik merambat dengan kecepatan suara. Dengan mengukur waktu yang dibutuhkan sinyal pantulan balik, maka dapat ditentukan jarak sensor dari sebuah objek.. Semakin lama waktu yang dibutuhkan gelombang umpan balik yang diterima oleh receiver maka semakin jauh jarak dari objek.

## 2. 5. Mikrokontroler Arduino

Arduino merupakan perangkat keras berupa papan sirkuit (circuit board) yang di dalamnya terdapat mikrokontroler. Arduino diproduksi oleh perusahaan asal Italia. Mikrokontroler adalah integrated circuit / IC yang dapat menjalankan berbagai macam program. Mikrokontroler dapat memproses data dengan cepat dan hanya membutuhkan daya yang kecil untuk menjalankannya. Hardware Arduino berbasis mikrokontroler ATMEGA8, ATMEGA168, ATMEGA328 dan ATMEGA1280 (yang terbaru ATMEGA2560). Processor yang digunakan adalah Atmel AVR dan menggunakan Bahasa pemrograman milik Arduino. Arduino tidak lagi memerlukan adanya chip. Ini dikarenakan didalamnya sudah memiliki yang namanya bootloader yang dapat memberitahukan program dari komputer. ([www.arduino.cc](http://www.arduino.cc)).

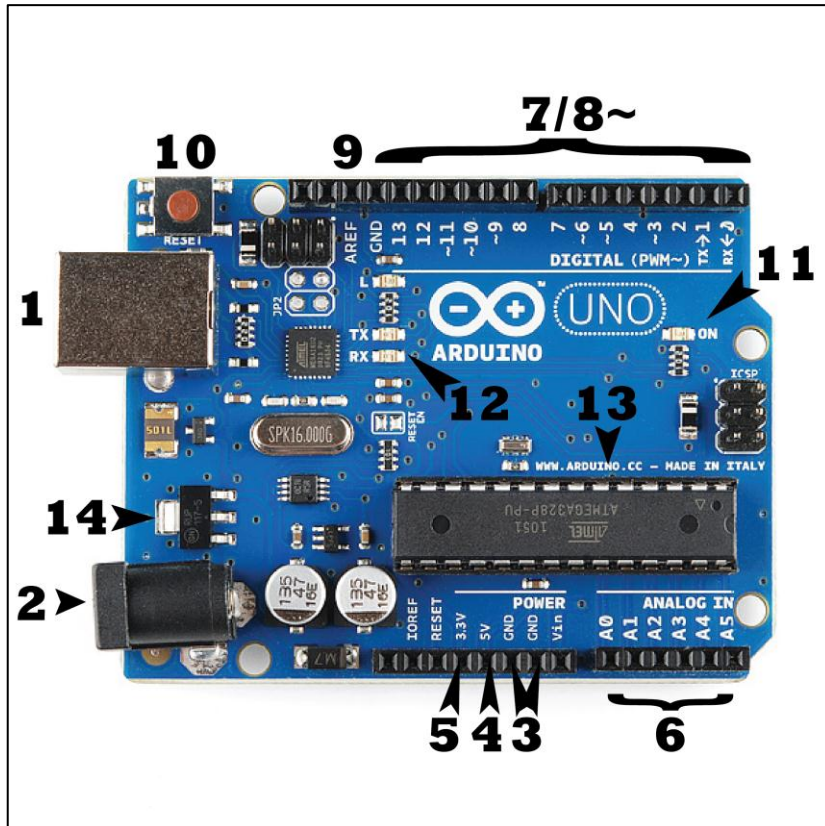


Gambar 2. 15 Mikrokontroler Arduino Uno  
Sumber : Elector.com

Terdapat berbagai macam Arduino semisal Arduino Uno dan Mega. Berbagai macam jenis Arduino dibuat menurut kecepatan pengolahan data. Hardware Arduino bersifat open source yang artinya desain dan spesifikasi circuit terbuka untuk umum, sehingga orang banyak (pihak ketiga) dapat memvariasikan desain arduino untuk kebutuhan dan fungsi spesifik tertentu.

Perusahaan Arduino juga memproduksi software yang dapat diunggah secara masal. Arduino Software Development merupakan perangkat lunak yang menghubungkan pengguna dengan hardware Arduino. Dalam pemrograman Arduino digunakan software Arduino IDE yang dapat digunakan pada system operasi Windows, Mac dan Linux. Pada umumnya dalam pemrograman mikrokontroler digunakan Bahasa pemrograman berupa kode binary dan data perintah diinput menggunakan hardware programming tertentu. Arduino banyak digunakan karena sederhana dan mudah untuk digunakan.

Dalam pemrograman, Arduino IDE digunakan Bahasa program Arduino ( Arduino Language Program) yang lebih mudah dan cepat untuk dipahami. Arduino IDE berbasis pada Bahasa pemrograman C programming. Program yg dimasukan pada Arduino IDE disebut sketch.



Gambar 2. 16 Bagian Bagian Arduino

Sumber : [learn.sparkfun.com](http://learn.sparkfun.com)

Bagian bagian yang ada dalam Arduino yaitu ([learn.sparkfun.com](http://learn.sparkfun.com)) :

- **(1) USB Port**  
Menghubungkan computer dan Arduino via kabel USB. Port ini juga bisa berguna sebagai suplai tenaga bagi Arduino dengan memakai arus dari computer.
- **(2) Power Jack**  
Mengubungkan Arduino dengan sumber daya
- **(3)(4)(5) Pin Ground, 5 V, dan 3.3 V**  
Pin ground berguna sebagai pin yang menghubungkan dengan ground pada sirkuit. Pin 5 V dan 3.3 V bergungsi untuk menyalurkan tegangan pada komponen.
- **(6) Pin Analog**  
Area yang terdiri dari label A0 sampai A5. Pin dapat membaca sinyal dari analog sensor seperti sensor ultrasonic dan mengubahnya menjadi sinyal digital. Arduino dapat membaca sinyal analog hingga 10 bit resolusi yang mana itu bernilai integer dari 0 sampai 1023.

- **(7) Pin Digital**  
Area yang terdiri dari label 0 sampai 13. Pin dapat mengirim ataupun menerima sinyal digital yang bernilai 0 (mati) dan 1 (hidup) .
- **(8) Pin PWM**  
Area pada pin digital yang terdiri dari label 3,5,6,9,10 dan 11. Pin dapat mengirim ataupun menerima sinyal digital dengan teknik PWM ( Pulse With Modulation).
- **(9) Analog Reference**  
Terkadang dapat digunakan sebagai sumber referensi voltase eksternal
- **(10) Tombol Reset**  
Dapat digunakan sebagai tombol reset untuk mengulang program dalam keadaan standby.
- **(11) Indikator LED**  
LED yang menjadi indikator bahwa Arduino bekerja dan terhubung dengan daya.
- **(12) TX RX LED**  
TX dan RX merupakan kepanjangan dari transmit dan receive. Digunakan dalam penggunaan transmisi komunikasi dengan komponen lain semisal Bluetooth.
- **(13) Integrated Circuit**  
Merupakan bagian utama dari mikrokontroler Arduino (ATMEGA) yang digunakan untuk mengolah data.
- **(14) Voltage Regulator**  
Mengontrol besaran voltase yang masuk ke Arduino.
- **Chip**  
Merupakan mikrokontroler sekunder yang digunakan untuk menghubungkan dan mengatur komunikasi antara USB dan Arduino board. Chip ini menghubungkan data secara bolak balik antara Arduino dan computer yang digunakan.
- **Crystal Resonator**  
Terhubung dengan mikrokontroler dan mengatur kecepatan mikrokontroler



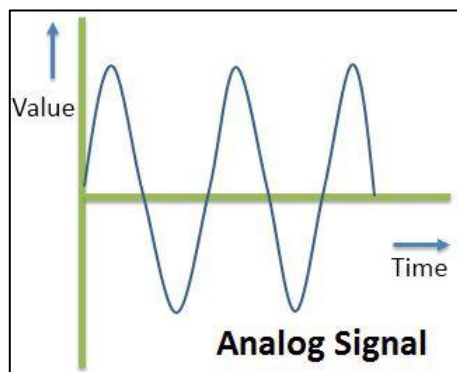
### 2.4.1 Sinyal Digital dan Analog

Terdapat dua jenis sinyal yang dapat ditransmisikan pada Arduino, yaitu sinyal digital dan sinyal analog.

*Tabel 2. 4 Perbedaan Sinyal Analog dan Digital*

Sumber : techdifferences.com

Perbedaan	Analog	Digital
Basic	Gelombang yang kontinu yang berubah tiap satuan waktu.	Gelombang diskrit yang membawa informasi dalam bentuk bilangan biner.
Bentuk	Berbentuk Gelombang Sinusoidal	Berbentuk Gelombang Kotak
Deskripsi	Dideskripsikan dengan amplitudo, periode, frekuensi dan fase.	Dideskripsikan dengan laju bit dan interval bit.
Rentang	Pada Arduino sinyal analog 10 bit resolusi yang mana itu bernilai integer dari 0 sampai 1023.	Pada Arduino nilai tetap sebesar 1 ( 5V/3.3V) atau 0 (0V)
Gangguan	Lebih Rentan terhadap distorsi	Kurang Rentan terhadap distorsi
Transmisi	Data dikirim berbentuk gelombang	Data dikirim berbentuk biner yaitu 1 atau 0



*Gambar 2. 17 Sinyal Analog*

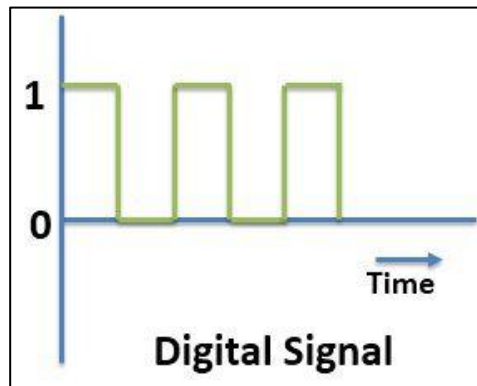
Sumber : techdifferences.com

Sinyal analog adalah bentuk gelombang kontinu yang berubah seiring berjalannya waktu. Sebuah sinyal analog lebih lanjut diklasifikasikan ke dalam sinyal sederhana dan komposit. Sinyal analog sederhana adalah gelombang sinus yang tidak dapat diurai lebih lanjut. Di sisi lain, sinyal analog komposit dapat diurai menjadi beberapa gelombang sinus. Sinyal analog dijelaskan dengan menggunakan amplitudo, periode atau frekuensi dan fase. Amplitudo menandai ketinggian maksimum sinyal.



Frekuensi menandai tingkat di mana sinyal berubah. Tahap menandai posisi gelombang sehubungan dengan waktu nol.

Sinyal analog tidak kebal terhadap kebisingan maka, itu menghadapi distorsi dan menurunkan kualitas transmisi. Kisaran nilai dalam sinyal analog tidak tetap. (techdifferences.com)



*Gambar 2. 18. Sinyal Digital*

Sumber : techdifferences.com

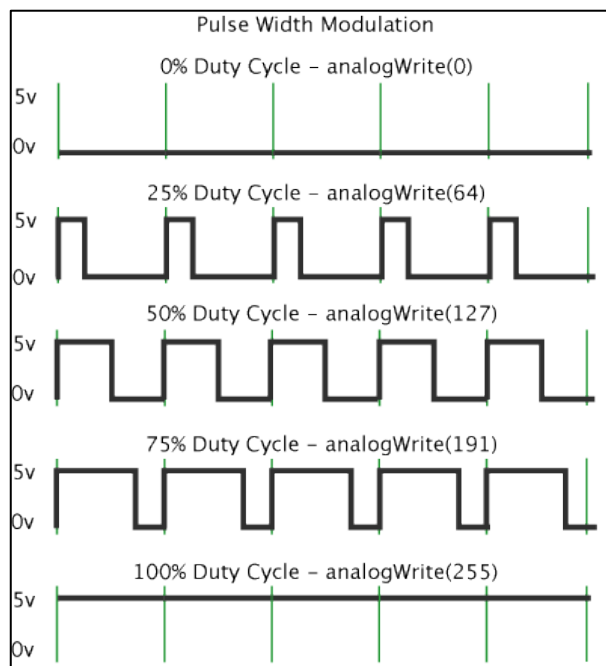
Sinyal digital juga membawa informasi seperti sinyal analog tetapi agak berbeda dari sinyal analog. Sinyal digital tidak kontinu, sinyal waktu diskrit. Sinyal digital mengandung informasi atau data dalam bentuk biner yaitu sinyal digital yang mewakili informasi dalam bentuk bit. Sinyal digital dapat lebih diurai menjadi gelombang sinus sederhana yang disebut harmonik. Setiap gelombang sederhana memiliki amplitudo yang berbeda, frekuensi dan fase. Sinyal digital dijelaskan dengan bit rate dan interval bit. Interval bit menjelaskan waktu yang dibutuhkan untuk mengirim sedikit. Di sisi lain, bit rate menggambarkan frekuensi interval bit.

Sebuah sinyal digital lebih kebal terhadap kebisingan; oleh karena itu, itu tidak menghadapi distorsi apapun. Sinyal digital lebih mudah untuk mengirimkan dan lebih dapat diandalkan bila dibandingkan dengan sinyal analog. Sinyal digital memiliki rentang nilai yang terbatas. Sinyal digital terdiri dari 0s dan 1s.

Sinyal analog merupakan gelombang kontinu yang terus berubah selama periode waktu. Di sisi lain, sinyal digital mewakili gelombang noncontinuous yang membawa informasi dalam format biner dan memiliki nilai diskrit. Sementara berbicara tentang sinyal analog kami menggambarkan perilaku gelombang dalam hal amplitudo, periode atau frekuensi, dan fase gelombang. Di sisi lain, sambil berbicara tentang sinyal diskrit kami menggambarkan perilaku gelombang dalam hal laju bit dan interval bit.

### 2.4.2 Pulse Width Modulation

*Pulse width modulation*, atau PWM, adalah teknik untuk mendapatkan hasil berupa sifat sinyal analog dengan menggunakan sinyal digital. Kontrol digital menghasilkan gelombang berbentuk persegi, di mana sinyal berubah antara keadaan ON dan OFF. Pola ON-OFF ini pada Arduino mensimulasikan tegangan penuh (5 volt) saat onn dan tegangan (0 volt) saat off . PWM dilakukan dengan mengubah porsi durasi waktu yang dihabiskan sinyal pada waktu ON dibandingkan dengan durasi waktu yang dihabiskan sinyal pada keadaan OFF. Durasi waktu tersebut disebut lebar pulsa (Pulse Width). Untuk mendapatkan nilai analog yang bervariasi, dilakukan dengan mengubah, atau memodulasi lebar pulsa/periode (Pulse Width). Sebagai contoh jika kita mengulangi pola on-off ini cukup cepat dengan LED misalnya, hasilnya adalah seolah-olah sinyal tegangan stabil antara 0 dan 5V dapat mengendalikan kecerahan LED.



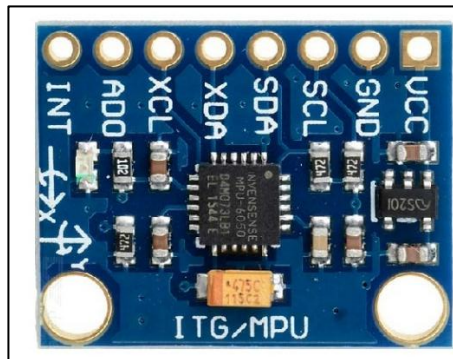
Gambar 2. 19 Grafik PWM

Sumber : Arduino.CC

Pada grafik di bawah ini, garis hijau mewakili periode waktu reguler. Durasi atau periode ini adalah kebalikan dari PWM frekuensi. Dengan kata lain, dengan frekuensi PWM Arduino di sekitar 500Hz, jarak garis hijau akan terukur sebesar 2 milidetik masing-masing. Pada Arduino nilai rentang sinyal PWM adalah 1 mili detik /1000 microsecond sampai 2 mili detik / 2000 microsecond.

## 2. 6. Gyro- Accelerometer

Sensro Gyro digunakan sebagai referensi arah (*heading*) sistem *Dynamic Positioning* pada model *supply vessel*. Sensor digunakan pada pengaturan pergerakan *Yaw* dimana kapal akan melakukan gerakan rotasi terhadap sumbu Z. Sensor akan mendeteksi besaran sudut yang dibentuk akibat pergerakan tersebut. Besaran sudut yang didapat kemudian diolah oleh mikrokontroler Arduino. Sensor yang dipakai pada tugas akhir ini MPU 6050. Sensor mempunyai 3 Axis Accelerometer dan 3 Axis Gyroscope yang terintegrasi pada chip.

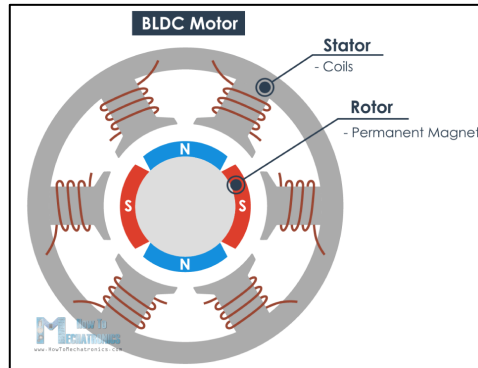


Gambar 2. 20 Gyro-Accelerometer  
Sumber :Amazon.com

Gyroscope mengukur kecepatan putar atau rata-rata perubahan sudut tiap satuan waktu pada sumbu X, Y, Z. Pembacaan dari Gyroscope berupa kecepatan sudut, atau besar sudut per satuan detik. Untuk mendapatkan sudut terukur maka kecepatan sudut harus diintegrasikan. Accelerometer mengukur akselerasi gravitasi pada tiap sumbu menggunakan trigonometri dengan mengukur sudut pada posisi sensor itu berada.

## 2.7. Brushless DC Motor

Merupakan jenis motor DC. Terdiri dari dua bagian yaitu stator dan rotor. Rotor merupakan bagian yang berputar adalah magnet permanen. Dan stator merupakan bagian yang diam terdiri dari susunan lilitan.



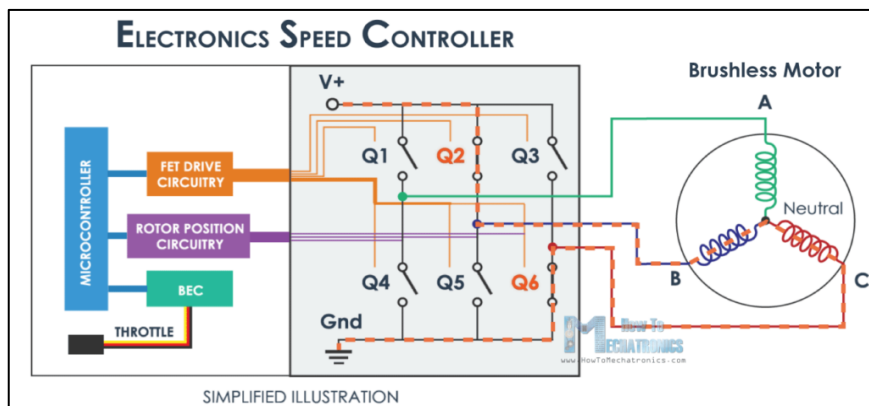
Gambar 2. 21. Bagian Motor

Sumber : [howtomechatronics.com](http://howtomechatronics.com)

Arus listrik disalurkan pada lilitan di bagian stator dan membangkitkan medan magnet yang memiliki kutub sesuai dengan arah arus. Medan magnet pada stator akan menggerakkan magnet permanen pada rotor. Jika lilitan diaktifkan satu persatu, maka rotor akan berputar akibat interaksi magnet. Putaran dari motor bisa diatur dengan cara mengatur arus yang masuk pada stator. Pengaturan arus dapat dilakukan menggunakan Electronic Speed Controller.

## 2.8. Electronic Speed Controller

ESC mengatur gerakan dan kecepatan dari motor DC dengan mengaktifkan MOSFET yang sesuai untuk menghasilkan medan magnet untuk memutar rotor. Semakin tinggi frekuensi arus yang dikirim semakin cepat gerakan putaran motor.



Gambar 2. 22. Rangkaian Motor

Sumber : [howtomechatronics.com](http://howtomechatronics.com)

## 2. 9. Kontrol PID

Kontroler adalah perangkat yang bertujuan untuk mengendalikan respon keluaran dari sebuah sistem. Input kontroler berupa sinyal dari sensor dan sinyal set point (masukan yang diinginkan). Proses yang terjadi pada kontroler adalah membandingkan kedua sinyal input, apakah sinyal dari sensor sudah sesuai dengan sinyal set point kemudian output kontroler berupa sinyal perintah yang dikirim pada aktuatur

PID (Proportional-Integral-Derivative) Controller adalah sebuah controller yang memiliki sistem umpan balik yang disebut *feedback*. *Feedback* atau umpan balik tersebut berperan sebagai pemberi informasi bahwa hasil output yang dikeluarkan oleh sistem belum sesuai dengan input yang diberikan. Dari umpan balik tersebut, sistem kontrol akan menghitung nilai *error* melakukan koreksi hingga pada akhirnya hasil output akan sesuai dengan input yang diberikan. Keunggulan PID dibandingkan dengan P dan PI adalah adanya *Derivative* yang akan mempengaruhi *overshoot* yang dihasilkan pada sebuah sistem. Metode yang digunakan pada PID adalah metode Ziegler-Nichols dimana metode tersebut akan memberikan hasil grafik yang cukup akurat. Pada PID berbasis Genetic Algorithm, optimalisasi dilakukan pada koefisien PID. Koefisien tersebut adalah nilai yang di dapat dari trial and error.

PID Fungsi dari PID sendiri dapat adalah sebagai berikut,

$$u(t) = K_p e(t) + K_I \int_0^t e(t) dt + K_D \frac{de(t)}{dt}$$

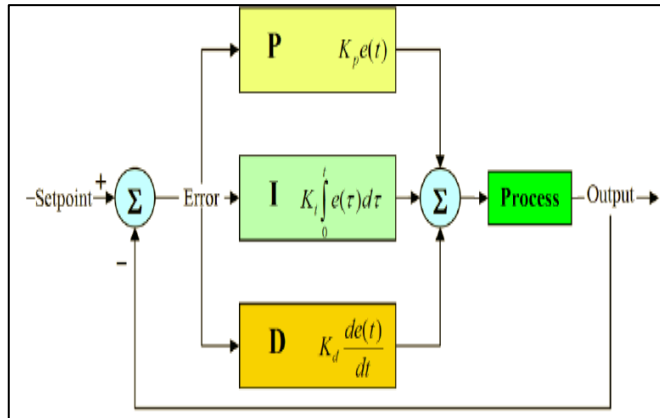
Gambar 2. 23 Fungsi PID

dimana terdapat tiga parameter utama, yaitu K<sub>p</sub> (Proportional), K<sub>i</sub> (Integral), dan K<sub>d</sub> (Derivative). Ketiga parameter tersebut memiliki fungsi masing-masing dalam respon sistem. Dengan mengatur ketiga parameter tersebut waktu respon yang didapat dapat berkurang, mengurangi nilai *error steady state*, dan mengurangi nilai overshoot yang terjadi. Pada tabel 2 dapat dilihat karakteristik kontroler PID.

Tabel 2. 5 Karakteristik PID

Controller Response	Rise Time	Overshoot	Settling Time	Steady State Error
K <sub>p</sub>	Decrease	Increase	Small Change	Decrease
K <sub>i</sub>	Decrease	Increase	Increase	Eliminate
K <sub>d</sub>	Small Change	Decrease	Decrease	Small Change

Gambar di bawah yang merupakan skema dari kontroler PID menunjukkan bahwa output akan dinilai kembali nilai errornya dan diatur dengan menggunakan kontroler PID sehingga didapatkan nilai koreksi yang selanjutnya akan digunakan pada proses dan menghasilkan nilai output yang memiliki nilai error sesuai dengan keinginan pemberi input.



Gambar 2. 24. Skema Kontroler PID

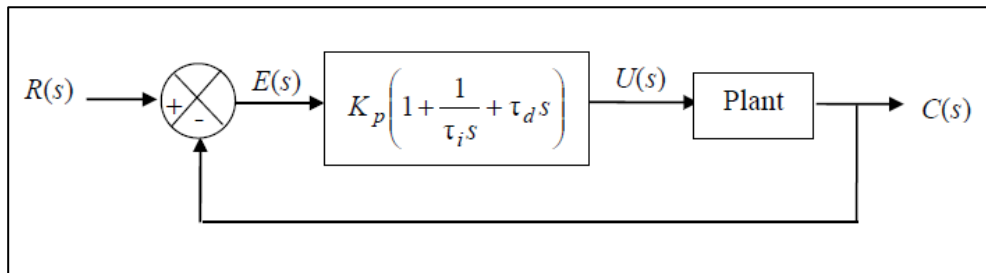
Pada kontroler PID juga perlu dilakukannya sebuah tuning dimana tuning tersebut dilakukan untuk mendapatkan nilai dari parameter dan selanjutnya digunakan untuk dibuat dalam bentuk grafik respon. Pada umumnya, tuning yang dilakukan pada kontroler PID adalah dengan menggunakan metode Zeigler and Nichols atau dapat berupa uji coba langsung pada percobaan. Berikut contoh ilustrasi dari perbandingan karakteristik PID

Control mode	Typical system responses	Advantages/disadvantages
On /off		<ul style="list-style-type: none"> <li>■ Inexpensive</li> <li>■ Simple</li> <li>■ Operating differential can be outside of process requirements</li> </ul>
Proportional P		<ul style="list-style-type: none"> <li>■ Simple and stable</li> <li>■ Fairly high initial deviation (unless a large P-band is chosen), then sustained offset</li> <li>■ Easy to set up</li> <li>■ Offset occurs</li> </ul>
Proportional plus Integral P + I		<ul style="list-style-type: none"> <li>■ No sustained offset</li> <li>■ Increase in proportional band usually required to overcome instability</li> <li>■ Possible increased overshoot on start-up</li> </ul>
Proportional plus Derivative P+D		<ul style="list-style-type: none"> <li>■ Stable</li> <li>■ Some offset</li> <li>■ Rapid response to changes</li> </ul>
Proportional plus Integral plus Derivative P+I+D		<ul style="list-style-type: none"> <li>■ Will give best control, no offset and minimal overshoot</li> <li>■ More complex to set up manually but most electronic controllers have an 'autotune' facility.</li> <li>■ More expensive where pneumatic controllers are concerned</li> </ul>

Gambar 2. 25 Ilustrasi Respon Kontroler

## 2. 10. Metode Zieger -Nichols

Terkadang pemodelan matematis suatu plant susah untuk dilakukan. Jika hal ini terjadi maka perancangan kontroler PID secara analitis tidak mungkin dilakukan sehingga perancangan kontroler PID harus dilakukan secara eksperimental. Pada bagian ini akan dibahas tentang perancangan kontroler PID secara eksperimental dengan menggunakan aturan Ziegler – Nichols. Ziegler dan Nichols memberikan aturan untuk menentukan nilai penguatan proporsional  $K_p$ , waktu integral  $\tau_i$ , dan waktu differensial  $\tau_d$  yang didasarkan pada karakteristik respon transien dari plant. Suatu kontroler PID yang diterapkan pada suatu plant akan menghasilkan sistem umpan balik yang hubungan antara masukan dan keluaran dinyatakan dalam diagram blok berikut ini :



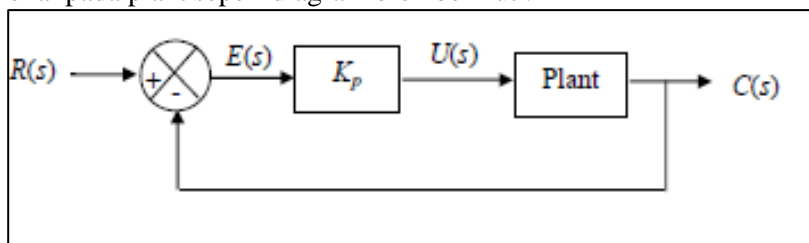
Gambar 2. 26 Diagram Blok PID

Dimana fungsi alih kontroler PID adalah :

$$\frac{U(s)}{E(s)} = K_p \left( 1 + \frac{1}{\tau_i s} + \tau_d s \right)$$

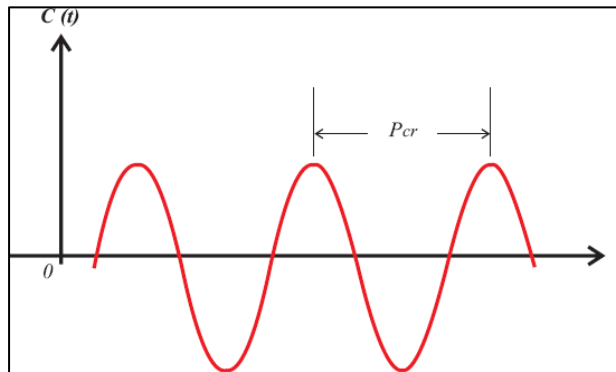
Gambar 2. 27 Fungsi Alih PID

Aturan Zieger & Nichols menentukan nilai parameter  $K_p$ ,  $\tau_i$ , dan  $\tau_d$  berdasarkan pada respon plant terhadap masukan sinyal step secara eksperimental atau berdasarkan pada nilai  $K_p$  yang dihasilkan dalam kestabilan marginal bila hanya aksi kendali proporsional yang digunakan. Ada dua metode penalaan Ziegler – Nichols yang bertujuan mencapai maksimum overshoot 25 %. Dalam metode kedua, mula – mula kita tentukan  $\tau_i = \infty$  dan  $\tau_d = 0$ . Dengan menerapkan kontroler proporsional pada plant seperti diagram blok berikut :



Gambar 2. 28 Diagram Blok PID

Kita atur nilai  $K_p$  dari nol ke suatu nilai kritis  $K_{cr}$ . Dalam hal ini, mula-mula keluaran plant memiliki osilasi yang berkesinambungan dengan periode  $P_{cr}$  seperti gambar berikut.



Gambar 2. 29 Osilasi PID

Jika keluaran tidak memiliki osilasi yang berkesinambungan untuk nilai  $K_p$  maka metode kedua aturan Ziegler-Nichols ini tidak berlaku.

Ziegler dan Nichols menyarankan penentuan nilai  $K_p$ ,  $\tau_i$ , dan  $\tau_d$  berdasarkan rumus yang diperlihatkan pada tabel berikut :

Tabel 2. 6 Tabel Permsamaan Siegler Nichols

Tipe kontroler	$K_p$	$\tau_i$	$\tau_d$
P	$0,5 K_{cr}$	$\infty$	0
PI	$0,45 K_{cr}$	$\frac{1}{P_{cr}}$ 1,2	0
PID	$0,6 K_{cr}$	$0,5 P_{cr}$	$0,125 P_{cr}$

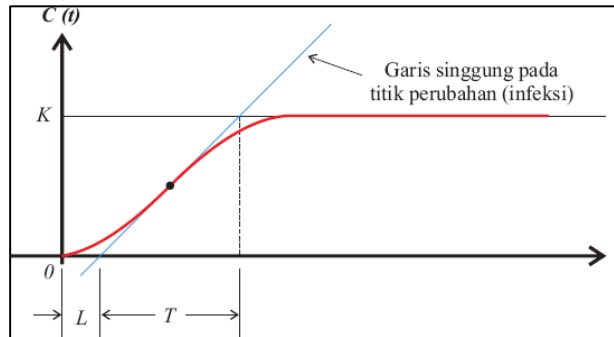
Jadi kontroler PID memiliki kutub pada titik asal dan dua nilai nol pada  $s = -4/P_{cr}$ . Fungsi alih kontroler PID berdasarkan metode kedua aturan Ziegler-Nichols adalah

$$\begin{aligned} \frac{U(s)}{E(s)} &= K_p \left( 1 + \frac{1}{\tau_i s} + \tau_d s \right) \\ &= 0,6 K_{cr} \left( 1 + \frac{1}{0,5 P_{cr} s} + 0,125 P_{cr} s \right) \\ \frac{U(s)}{E(s)} &= 0,075 K_{cr} P_{cr} \frac{\left( s + \frac{4}{P_{cr}} \right)^2}{s} \end{aligned}$$

Gambar 2. 30 Fungsi Alih kontrollerr



Dalam metode lain, kita perlu mendapatkan respon plant terhadap masukan sinyal step. Jika plant tidak mengandung integrator atau kutub pasangan kompleks yang dominan, maka kurva respon step plant tersebut kelihatan seperti kurva bentuk S.. Jika respon plant tidak memiliki kurva berbentuk S, metode ini tidak berlaku. Kurva respon step dapat dihasilkan secara eksperimen atau dari simulasi dinamik sistem. Kurva respon step berbentuk S dapat kita lihat seperti berikut ini :



Gambar 2. 31 Kurva Respon

Kurva berbentuk S dikarakteristikan oleh dua parameter yaitu waktu tunda  $L$  dan konstanta waktu tunda  $T$ . Konstanta waktu tunda  $T$  ditentukan dengan menggambar garis singgung pada titik perubahan kurva berbentuk S dan menentukan perpotongan garis singgung dengan sumbu waktu dan garis  $c(t) = K$ . Fungsi alih loop tertutup plant dengan kurva respon step berbentuk S ini dapat didekati dengan system orde pertama dengan ketrlambatan transport, yaitu :

$$\frac{C(s)}{U(s)} = \frac{Ke^{-Ls}}{Ts + 1}$$

Gambar 2. 32 Persamaan Kurva Respon

Ziegler dan Nichols menyarankan penentuan nilai  $K_p$ ,  $\tau_i$ , dan  $\tau_d$  berdasarkan rumus yang diperlihatkan pada tabel berikut :

Tabel 2. 7 Penentuan Koefisien

Type kontroler	$K_p$	$\tau_i$	$\tau_d$
P	$\frac{T}{L}$	$\square$	0
PI	$0,9 \frac{T}{L}$	$L$	0
PID	$1,2 \frac{T}{L}$	$2L$	0,5L

“ Halaman ini sengaja dikosongkan”

## **BAB III METODOLOGI**

Pada Bab ini akan menguraikan langkah langkah sistematis yang dilakukan dalam penelitian. Metodologi merupakan kerangka dasar dari tahapan penyelesaian tugas akhir. Metodologi mencakup semua kegiatan yang dilaksanakan untuk menjawab rumusan masalah atau melakukan analisa terhadap percobaan yang dilakukan. Metodologi dijadikan acuan untuk menyelesaikan penelitian tugas akhir tentang rancang bangun Dynamic Positioning System pada model Supply Vessel menggunakan Sensor ultrasonik dengan kontrol Arduino Berbasis PID.

### **3. 1. Perumusan Masalah**

Tahapan awal dari penyelesaian tugas akhir ini adalah perumusan masalah. Pada tahap ini dilakukan pengidentifikasian masalah yang akan dihadapi dalam rancang bangun alat, serta masalah yang mungkin terjadi. Penyusunan latar belakang, rumusan dan batasan masalah dibuat agar penulisan lebih focus dan mencapai hasil yang maksimal. Perumusan masalah telah dituliskan pada bab 1 dalam rumusan masalah. Pada tugas akhir ini permasalahan yang ada seperti bagaimana mendesain dan merancang system sensorik Dynamic positioning pada model supply vessel dengan Kontrol Arduino dan merancang program kendali Dynamic positioning dengan Kontrol Arduino berbasis PID . Setelah identifikasi masalah, dilakukan studi literature yang bertujuan untuk mengumpulkan informasi pendukung terkait penulisan tugas akhir.

### **3. 2. Studi Literatur**

Pada tahap ini dilakukan pencarian preferensi, dasar dasar informasi dan pembelajaran dari literatur-literatur yang berhubungan dengan penelitian untuk membantu penulis dalam mendesain alat yang akan dibuat dan mendapatkan hasil percobaan yang ideal. Literatur-literatur yang digunakan dapat dicari dari berbagai media anatara lain situs internet, buku, jurnal nasional maupun jurnal internasional, paper nasional maupun paper internasional , hasil diskusi dan bimbingan dengan dosen, serta pembelajaran di luar kuliah dan lain lain. Adapun capaian studi yang direncanakan mencakup :

- Studi Literatur mengenai karakteristik Dynamic Positioning System secara umum dan pengaplikasiannya pada supply vessel
- Studi Literatur mengenai karakteristik mikrokontroler Arduino
- Studi Literatur mengenai karakteristik dari peralatan sensor dan komponen pendukung lain yang akan digunakan dalam pembuatan alat
- Studi literatur mengenai karakteristik dari sistem kontrol PID
- Studi literatur mengenai metode pemrograman pada Arduino berbasis kontrol PID

### 3.3. Desain dan Perancangan Alat

Pada tahap ini dibuat perancangan desain dan perencanaan alat dan bahan yang akan dipasang pada protipe model supply vessel. Perancangan dan perencanaan juga didasari oleh pertimbangan terhadap studi literature yang dilakukan di tahap sebelumnya. Seiring dengan berjalannya tahapan selanjutnya, hasil desain dan rancangan yang ada akan dilakukan perbaikan ataupun perubahan sesuai dengan pembaharuan dan kendala dalam percobaan yang akan dilakukan. Desain dan Rancangan yang dibuat mencakup :

### 3.4. Desain dan Rancangan Komponen Sistem Mekanik

Desain dan rancangan komponen sistem mekanik mencakup sistem perancangan motor penggerak yang akan dipasang pada protipe model supply vessel .

#### 3.3.1 Desain dan Rancangan Komponen Sistem Elektronik

Desain dan rancangan komponen sistem elektronik mencakup desain rangkaian elektronik dan wiring diagram dari sirkuit elektronik yang akan dipasang pada protipe model supply vessel .

#### 3.3.2 Alat dan Bahan

Perkiraan peralatan dan bahan yang direncanakan untuk digunakan pada tugas akhir ini mencakup :

#### 1) Protipe Kapal Supply Vessel

Model kapal yang akan digunakan sebagai model dynamic positioning system dan akan dilakukan pengujian adalah jenis kapal supply vessel. Berikut Dimensi kapal yang akan digunakan :

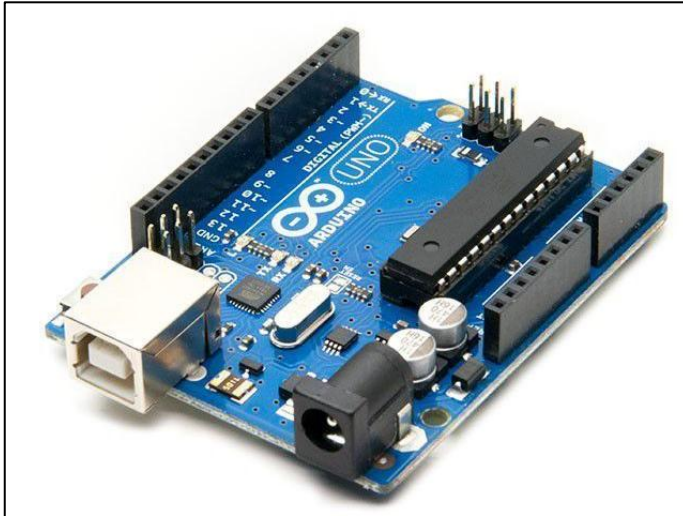
Tabel 3. 1. Dimensi Utama Kapal

Parameter	Kapal Asli	Kapal Model
Length Overall	26.70 m	
LBP	22.95 m	1.02 m
Height		0.9 m
Draught	1.4 m	0.5 m
Beam Hull ( Hull)	6.00 m	
Beam ( Overall)	6.50 m	0.3 m
Fuel Capacity	17.0 m <sup>3</sup>	
Fresh Water Capacity	5.0 m <sup>3</sup>	



## 2) Mikrokontroler Arduino

Arduino yang dipakai merupakan jenis Arduino Uno. Arduino merupakan mikrokontroler yang berfungsi mendeteksi dan menerima data hasil pembacaan sensor ultrasonic dan *Gyro-Accelerometer* yang nantinya mengolah data untuk kemudian mengirim perintah berbasis PID kepada motor pendorong (*Thruster*) berupa putaran baling baling (*Propeller*).

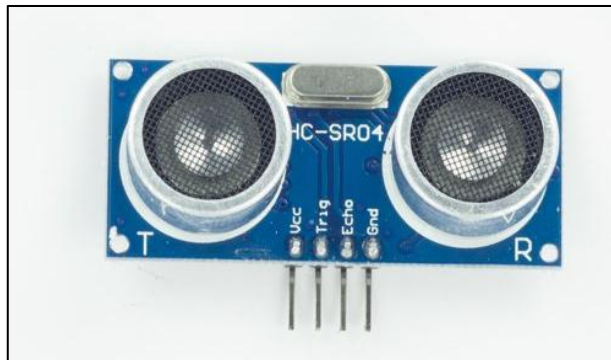


Gambar 3. 3 Arduino

Sumber : Elector.com

## 3) Sensor Ultrasonik

Sensor Ultrasonik digunakan sebagai referensi posisi sistem *Dynamic Positionig* pada model supply vessel. Sensor akan mengirimkan sinyal ultrasonic yang kemudian akan dipantulkan oleh bangunan rig kembali ke sensor. Bagian sensor akan mendeteksi jarak *supply vessel* terhadap rig dan besaran jarak bisa didapat dan diolah oleh mikrokontroler Arduino.

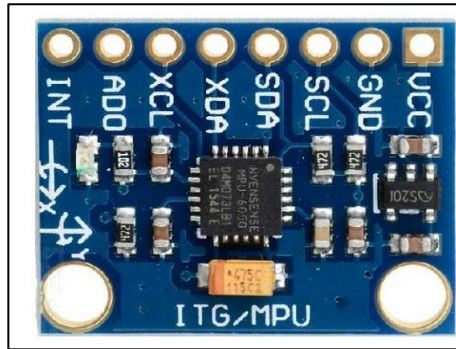


Gambar 3. 4 Sensor Ultrasonik

Sumber : Amazon.in

#### 4) Sensor Gyro-Accelormeter

Sensro Gyro digunakan sebagai referensi arah (*heading*) sistem *Dynamic Positioning* pada model *supply vessel*. Sensor digunakan pada pengaturan pergerakan *Yaw* dimana kapal akan melakukan gerakan rotasi terhadap sumbu Z. Sensor akan mendeteksi besaran sudut yang dibentuk akibat pergerakan tersebut. Besaran sudut yang didapat kemudian diolah oleh mikrokontroler Arduino.



Gambar 3. 5 Gyro Accelerometer  
Sumber :Amazon.com

#### 5) DC Brushless Motor

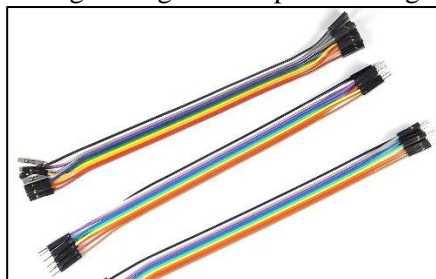
Motor Digunakan sebagai penggerak. Motor memberikan gaya kepada kapal sesuai perintah dari mikrokontroler. Perubahan gaya yang dihasilkan merupakan akibat dari pengaturan putaran propeller.



Gambar 3. 6 Motor DC  
Sumber :Aliexpress

#### 6) Kabel

Kabel digunakan untuk menghubungkan komponen sebagai pengalir arus listrik.



Gambar 3. 7Kabel  
Sumber :Aliexpress



### 7) **Electronic Speed Controller**

ESC mengatur gerakan dan kecepatan dari motor DC dengan mengatur suplai arus yang masuk ke dalam motor. Pengaturan arus sesuai dengan perintah dari mikrokontroler.



*Gambar 3. 8 ESC*

Sumber : RC shop bd.com

### 8) **Servo**

Servo digunakan untuk memberi arah putaran pada motor DC. Servo dihubungkan dengan poros terhadap motor.



*Gambar 3. 9 Servo*

Sumber : Robu.in

### 9) **Baterai**

Baterai digunakan sebagai sumber listrik bagi motor. Baterai yang dipakai jenis Lithium Polymer ( LiPo).



*Gambar 3. 10 Baterai*

Sumber : Robomart.com



### 3. 5. Pembuatan Alat

Pada tahap ini dilakukan pembuatan dan perangkaian komponen sesuai dengan desain mekanik dan elektronik dari sistem.

### 3. 6. Pemograman

Pada tahapan ini dilakukan pemograman dengan memasukkan input perintah pada software Arduino IDE. Program dibuat berbasis PID untuk mencapai kemampuan Dynamic Positioning yaitu kapal dapat mampu mempertahankan posisi pada kecepatan ombak yang berbeda beda dari arah yang berbeda.

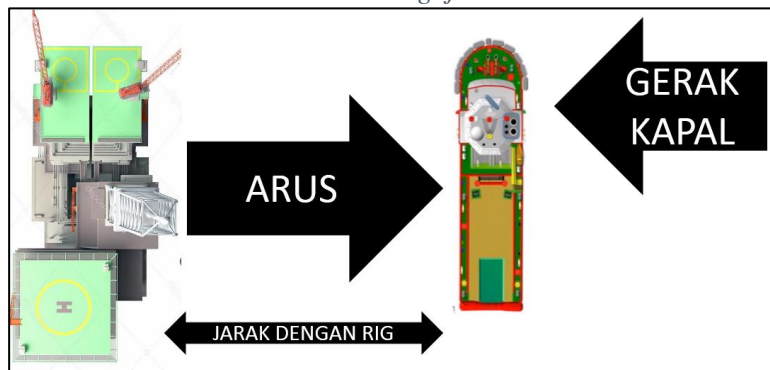
### 3. 7. Percobaan Alat

Pada tahapan ini pertama dilakukan trial dan error pada kapal model. Tahapan ini bertujuan sebagai tahapan tuning untuk mendapatkan variabel variabel PID pada setiap keadaan uji dan mendapatkan nilai putaran minimal untuk mempertahankan posisi diam. Setelah trial dan error dilakukan, maka akan dilakukan pengujian final dari model. Parameter yang dijadikan acuan adalah keberhasilan model kapal untuk mempertahankan posisi diam pada posisi dan kecepatan arus tertentu yang diberikan. Jika Hasil Trial dan Error tidak memenuhi keadaan ideal maka akan dilakukan identifikasi program dan rangkaian komponen, dan dilakukan tahapan pemograman ulang. Jika hasil sudah memenuhi keadaan ideal maka dilanjutkan pada tahapan kesimpulan..

Berikut ilustrasi dari pengujian 1 dan 2



Gambar 3. 11 Pengujian 1



Gambar 3. 12 Pengujian 2

Pada pengujian 1 arus datang dari sebelah Portside kapal. Dengan adanya arus maka kapal akan bergeser mendekati rig. Kuat arus yang keluar pada sumber berubah seiring dengan posisi pengujian dari sumber arus. Semakin jauh titik tertentu dari sumber arus maka arus akan semakin melemah. Berdasarkan pengukuran kecepatan arus pada titik keluar air dari sumber sebesar 0.010 m/s dan pada titik terjauh sebesar 0.001 m/s. Pengujian dilakukan pada tiap tiap jarak kapal dari rig yang ditentukan. Dengan nilai estimasi jarak, maka kapal harus berada pada titik tertentu sebesar jarak tertentu dari rig. Jarak itu sebesar 20 cm, 40 cm, 60 cm, 80 cm, 100 cm. Pengujian dilakukan dari kondisi error maksimal, yaitu saat kapal berada jauh dari titik referensi yaitu saat kapal berada sedekat mungkin di depan rig.

Pada pengujian 2 arus datang dari sebelah Portside kapal. Dengan adanya arus maka kapal akan bergeser menjauhi rig. Kuat arus yang keluar pada sumber berubah seiring dengan posisi pengujian dari sumber arus. Semakin jauh titik tertentu dari sumber arus maka arus akan semakin melemah. Berdasarkan pengukuran kecepatan arus pada titik keluar air dari sumber sebesar 0.010 m/s dan pada titik terjauh sebesar 0.001 m/s. Pengujian dilakukan pada tiap tiap jarak kapal dari rig yang ditentukan. Dengan nilai estimasi jarak, maka kapal harus berada pada titik tertentu sebesar jarak tertentu dari rig. Jarak itu sebesar 20 cm, 40 cm, 60 cm, 80 cm, 100 cm. Pengujian dilakukan dari kondisi error maksimal, yaitu saat kapal berada jauh dari titik referensi yaitu saat kapal berada sejauh mungkin dari rig.

### **3. 8. Analisan Data Percobaan dan Pembahasan**

Hasil yang diperoleh kemudian akan dipakai untuk menganalisa program yang ada. Jika sistem dan model dapat mempertahankan posisi terhadap perubahan keadaan, maka tugas akhir dikatan berhasil. Data yang didapat berupa hasil tuning PID yang ideal pada hasil percobaan dan respon putaran motor pada tiap keadaan percobaan.

### **3. 9. Kesimpulan dan Saran**

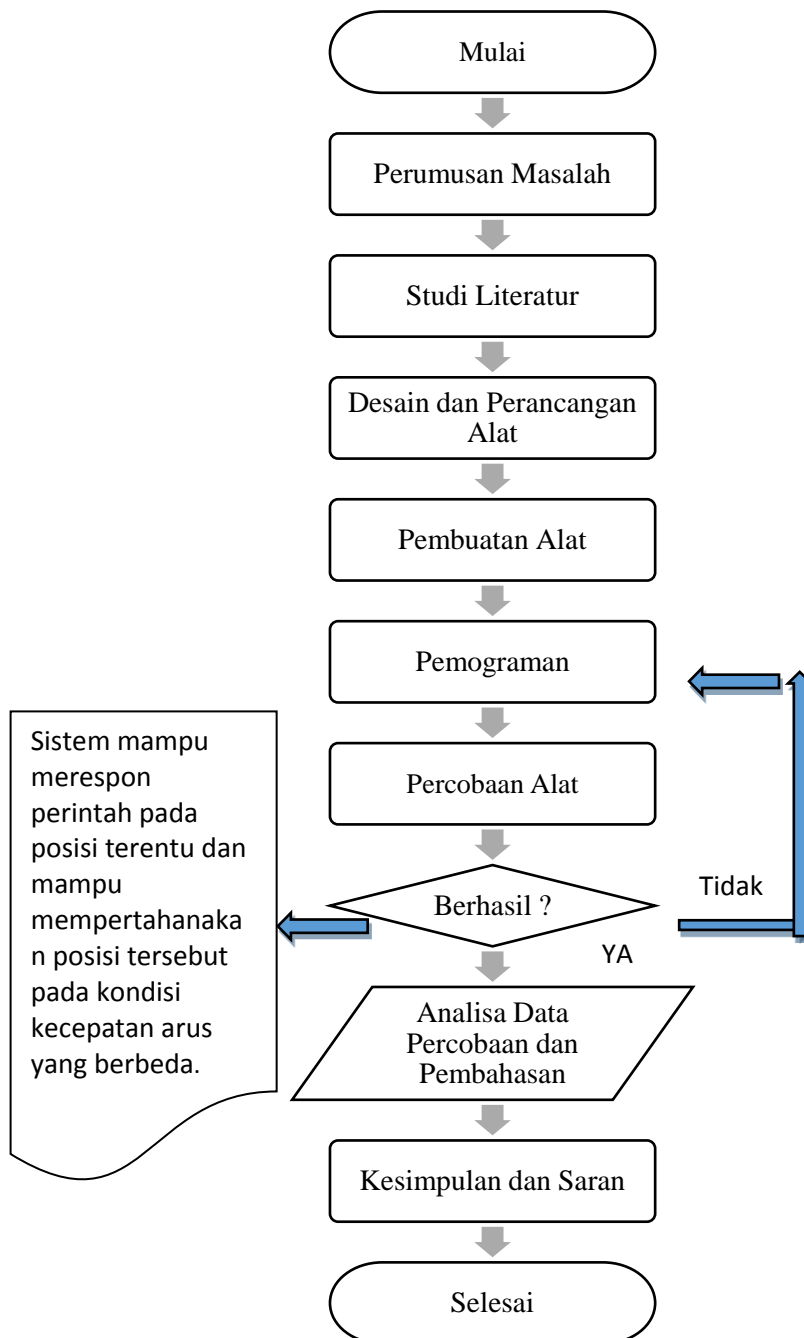
Langkah Terakhir adalah membuat kesimpulan keseluruhan proses yang telah dilakukan sebelumnya serta memberikan jawaban atas permasalahan yang ada, saran saran diberikan berdasarkan hasil dari penyusunan laporan yang dapat dijadikan dasar pada penelitian selanjutnya, baik terkait secara langsung pada penelitian ini ataupun pada data data dan metodologi yang nantinya akan direfrensi. Kesimpulan juga akan membahas secara umum bagaimana sistem bekerja.

### **3. 10. Penyusunan Laporan**

Melalui perolehan data pengujian dan perumusan yang didapat sebelumnya, maka pada dilakukan penyusunan laporan dari rumusan hingga kesimpulan dan saran guna untuk memudahkan untuk membaca tugas akhir ini.

### 3. 11. Flow Chart

Adapun Flow chart dari tugas akhir ini adalah sebagai berikut :



“ Halaman ini sengaja dikosongkan”

## BAB IV HASIL DAN PEMBAHASAN

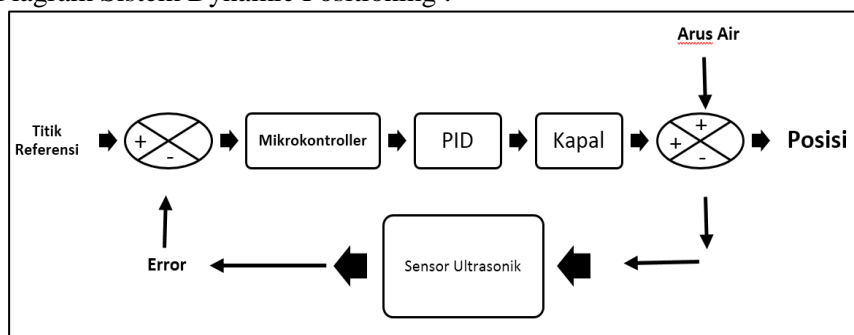
### 4.1. Gambaran Umum

Berikut komponen Sistem Dynamic Positioning yang digunakan pada tugas akhir . Lebih lanjut spesifikasi komponen akan dijelaskan di bagian peralatan.

Tabel 4. 1 Komponen DP System

Sub Sistem Dynamic Positioning	Komponen Model
Model :	Model Kapal Supply Vessel
Sistem Referensi Referensi Posisi ( Position Reference ) :	Sensor Ultrasonik SRF HC 04
Sistem Referensi Referensi Arah ( Heading Reference) :	Sensor Gyro Accelerometer MPU 6050
Sistem Kontrol :	Mikrokontroler Arduino UNO dengan pemograman berbasis PID
Sistem Penggerak :	Electronic Brushless DC Motor
Sistem Daya :	Baterai Lithium Polymer

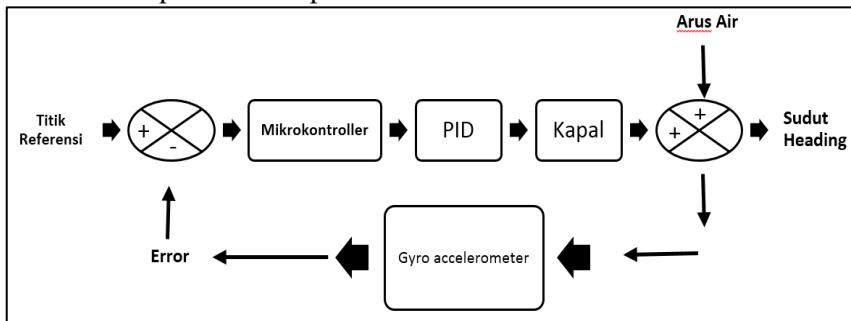
Sistem kontrol akan mengendalikan respon keluaran dari sistem penggerak. Input Sistem kontrol berupa sinyal dari Sistem Referensi yang kemudian akan dimasukkan nilai set point (masukan yang diinginkan). Proses yang terjadi pada Sistem kontrol adalah sistem akan membandingkan kedua nilai input, perbedaan nilai akan menjadi nilai error. Sistem kontrol akan mengolah data berbasis PID dan mengirimkan perintah untuk mengeliminir nilai error. Perintah dikirim dan dijalankan oleh sistem penggerak. Berikut Blok Diagram Sistem Dynamic Positioning :



Gambar 4. 1 Blok Diagram Sistem

Titik referensi dimasukkan ke dalam kontroler. Kemudian motor akan merespon dan bergerak menuju titik referensi yang dituju pada heading tertentu. Jarak dan sudut yang sebenarnya akan diukur oleh sensor. Kemudian nilai tersebut dibandingkan dengan nilai estimasi yang diinginkan. Selisih dari nilai tersebut akan dinyatakan sebagai error. Besaran error akan dieliminir oleh Arduino kemudian memberi perintah kepada motor penggerak. Pergerakan motor akan mengkoreksi nilai posisi pada keadaan sebenarnya

dan menghasilkan nilai error baru untuk diolah oleh kontroller. Proses terus berlanjut sampai error yang didapatkan sekecil mungkin dan putaran motor mencapai putaran minimal untuk mempertahankan posisi.



Gambar 4. 2 Blok Diagram Sistem 2

#### 4. 2. Respon yang diharapkan

Pada Tugas akhir ini model diuji pada kolam arus dengan diberi kecepatan tertentu secara kontinyu. Arus akan menyebabkan pergerakan pada model. Respon sistem yang diharapkan yaitu :

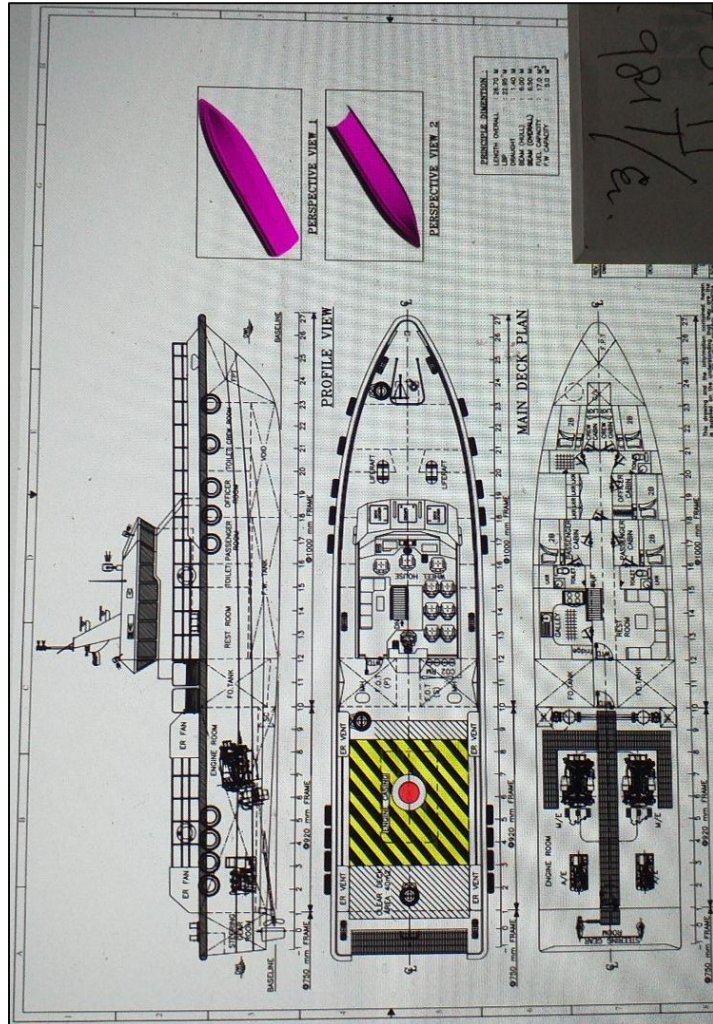
- Mampu Mendeteksi nilai error akibat perpindahan, kemudian nilai error akan diolah melalui PID dan menghasilkan nilai PWM berupa keluaran kecepatan motor. Yang menyebabkan terjadinya pergerakan koreksi posisi kapal.
- Jika tidak ada nilai error akibat perpindahan, motor akan beroperasi pada kecepatan minimum untuk mempertahankan posisi. Nilai tersebut didapat berdasar trial dan error sebelum pengujian final.
- Propeller akan berputar lebih cepat jika nilai error semakin besar dan lebih lambat jika nilai error semakin kecil sampai nilai error menjadi 0 dan motor beroperasi pada kecepatan normal.

#### 4. 3. Data Model Kapal

Model kapal yang akan digunakan sebagai model dynamic positioning system dan akan dilakukan pengujian adalah jenis kapal supply vessel. Berikut Dimensi kapal yang akan digunakan :

Tabel 4. 2 Dimensi Utama Kapal

Parameter	Kapal Asli	Kapal Model
Length Overall	26.70 m	
LBP	22.95 m	1.02 m
Height		0.9 m
Draught	1.4 m	0.5 m
Beam Hull ( Hull)	6.00 m	
Beam ( Overall)	6.50 m	0.3 m
Fuel Capacity	17.0 m <sup>3</sup>	
Fresh Water Capacity	5.0 m <sup>3</sup>	



Gambar 4. 3 General Arrangement



Gambar 4. 4 Model Kapal

#### 4. 4. Perancangan Sistem Elektronik

##### 4.3.1 Peralatan dan bahan

###### 1) Mikrokontroler Arduino

Pada tugas akhir ini digunakan mikrokontroller Arduino Uno. Arduino yang dipakai merupakan jenis Arduino Uno. Arduino merupakan mikrokontroller yang berfungsi mendeteksi dan menerima data hasil pembacaan sensor ultrasonic dan Gyro-Accelerometer yang nantinya mengolah data untuk kemudian mengirim perintah berbasis PID kepada motor pendorong ( Thruster) berupa putaran baling baling ( Propeller) . Berikut dijelaskan daftar pin dan koneksi pada komponen :

*Tabel 4. 3 Pin Arduino*

Pin	Koneksi
Ground Parallel	Ground SRF 1 Ground SRF 2 Ground SRF 3 Ground Motor 1 Ground Motor 2 Ground Servo 1 Ground Servo 2 Ground Gyro Accelerometer
VCC ( 5v)	VCC SRF 1 VCC SRF 2 VCC SRF 3 VCC Servo 1 VCC Servo 2 VCC Gyro Accelerometer
Analog A5	Echo SRF 1
Analog A4	Trigger SRF 1
Analog A3	Echo SRF 2/Gyro
Analog A2	Trigger SRF 2/Gyro
Analog A1	Echo SRF 3
Analog A0	Trigger SRF 3
Digital 5	Data Motor 1
Digital 6	Data Motor 2
Digital 10	Data Servo 1
Digital 11	Data Servo 2
Data pin	Komputer



## 2) SRF HC -04

Pada tugas akhir ini digunakan Sensor SRF HC-04. Sensor sebagai referensi posisi sistem *Dynamic Positioning* pada model supply vessel. Sensor akan mengirimkan sinyal ultrasonic yang kemudian akan dipantulkan oleh bangunan rig kembali ke sensor. Terdapat 3 buah sensor ultrasonic yang dipakai. Berikut spesifikasi sensor :

*Tabel 4. 4 Spesifikasi SRF*

Spesifikasi	Keterangan
Voltase	DC 5V
Arus	15- 20 mA
Frekuensi	40 Khz
Jarak Maksimum	4 m
Jarak Minimum	2 cm
Sudut ukur	15 derajat
Trigger Input Signal	10uS TTL pulse
Echo Output Signal	Input signal
Working Temperatur	-15°C – 70 °C
Dimensi	P = 43 mm , L = 20 mm, T= 15 mm
Diameter Lubang	1mm (M1)
Diameter Transmitter	8 mm

Berikut merupakan daftar pin yang digunakan pada SRF HC-04

*Tabel 4. 5 Pin SRF*

Pin	Keterangan
VCC	Pin Daya 5V
TRIG	Pin masukan sinyal, pemberi gelombang ( Pulse)
ECHO	Pin keluaran sinyal , penerima gelombang ( Pulse)
Ground	Pin Ground

### 3) Sensor Gyro-Accelormeter

Sensro Gyro digunakan sebagai referensi arah (*heading*) sistem *Dynamic Positioning* pada model *supply vessel*. Sensor digunakan pada pengaturan pergerakan *Yaw* dimana kapal akan melakukan gerakan rotasi terhadap sumbu Z. Sensor akan mendeteksi besaran sudut yang dibentuk akibat pergerakan tersebut. Besaran sudut yang didapat kemudian diolah oleh mikrokontroler Arduino. Sensor yang dipakai pada tugas akhir ini MPU 6050. Berikut Spesifikasi Sensor :

*Tabel 4. 6 Spesifikasi Gyro- Accelerometer*

Sumber : [ww.sunrom.com](http://ww.sunrom.com)

Spesifikasi	Keterangan
Chip	MPU-6050
Power supply	3~5V Onboard regulator
Communication mode	Standard IIC communication protocol
Gyroscopes range	+/- 250 500 1000 2000 degree/sec
Acceleration range	+/- 2g, +/- 4g, +/- 8g, +/- 16g
Pin pitch	2.54mm
Chip built-in 16bit AD converter, 16-bit data output	

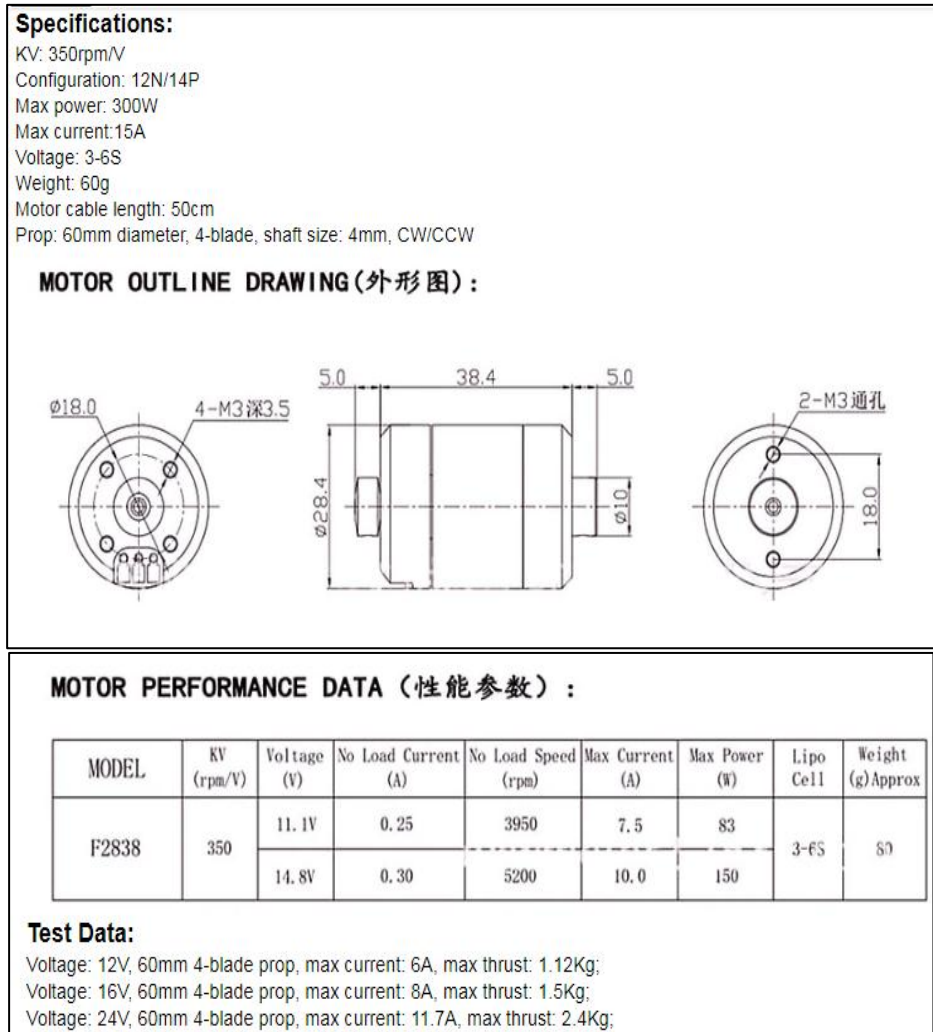
Berikut merupakan daftar pin yang digunakan pada MPU 6050 :

*Tabel 4. 7 Pin Gryro*

Pin	Keterangan
VCC	Pin Daya 5V
SDA	Pin masukkan sinyal sumbu
SCL	Pin masukkan sinyal sumbu
Ground	Pin Ground

#### 4) DC Brushless Motor

Motor Digunakan sebagai penggerak. Motor memberikan gaya kepada kapal sesuai perintah dari mikrokontroller. Perubahan gaya yang dihasilkan merupakan akibat dari pengaturan putaran propeller. Berikut spesifikasi dari motor :



Gambar 4. 5 Spesifikasi Motor

Sumber : [www.aliexpress.com](http://www.aliexpress.com)

Kv rating pada brushless motor menentukan jumlah RPM yang dapat dicapai motor pada tiap Volt saat tanpa beban. Berdasarkan tabel spesifikasi maka dengan penggunaan baterai 4 sel sebesar 14.8 V, maka kecepatan maximum motor yang dapat dicapai saat tanpa beban adalah 5200 rpm.

### 5) Electronic Speed Controller

ESC memiliki 3 kabel control fase untuk dihubungkan ke motor pada satu sisi. Di sisi lain ada 2 kabel VCC dan GND untuk aliran daya pada motor yang berasal dari baterai. Kabel lain adalah kabel sinyal untuk inputan pengaturan sinyal dari Arduino. Berdasarkan spesifikasi motor, maka ESC yang digunakan memiliki arus maksimal besar 30 A.

*Tabel 4. 8 Spesifikasi ESC*

Sumber : [www.aliexpress.com](http://www.aliexpress.com)

Merk	SimonK
Arus Konstan	30 A
Arus Maksimal	35 A
BEC Model	Linier Mode
BEC output	5 V 3 A
Berat	25 g
Baterai	LiPo 2-4 S Ni-MH Ni-Cd : 4-12 NIMH

### 6) Servo

Servo memiliki 3 kabel pada satu sisi. 2 kabel merupakan kabel power untuk ground dan voltase sedangkan kabel lain merupakan kabel inputan sinyal. Berdasarkan spesifikasi, servo menggunakan voltase yang diambil dari Arduino sebesar 5 V. Spesifikasi Servo yang dipakai sebagai berikut.

*Tabel 4. 9 Spesifikasi Servo*

Sumber : [servodatabase.com](http://servodatabase.com)

Merk	MG995
Modulasi	Digital
Torsi	4.8V: 130.54 oz-in (9.40 kg-cm) 6.0V: 152.76 oz-in (11.00 kg-cm)
Speed	4.8V: 0.20 sec/60° 6.0V: 0.16 sec/60°
Berat	1.94 oz (55.0 g)
Stal Current	2.5 A
Dimensi	Length:1.60 in (40.7 mm) Width:0.78 in (19.7 mm) Height:1.69 in (42.9 mm)
Running Current	500-900mA
Voltage	4.8-6 V
Tipe Gear	Metal

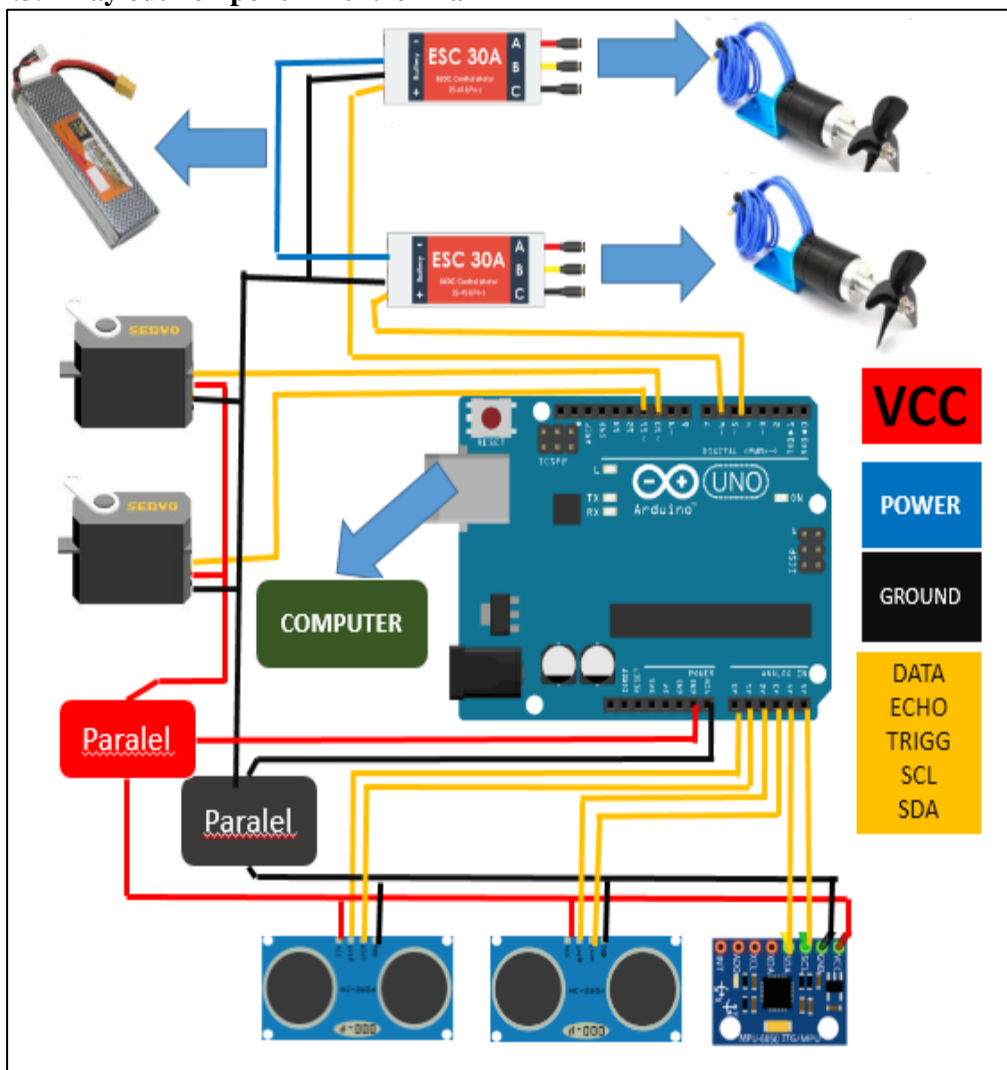
## 7) Baterai

Jenis motor yang digunakan yaitu motor DC brushless merupakan jenis motor yang cenderung boros daya sehingga membutuhkan daya baterai yang besar, yang mana pada umumnya digunakan baterai jenis Lihium Polymer yang lebih efisien dibanding baterai jenis lain. Berikut spesifikasi baterai yang digunakan.

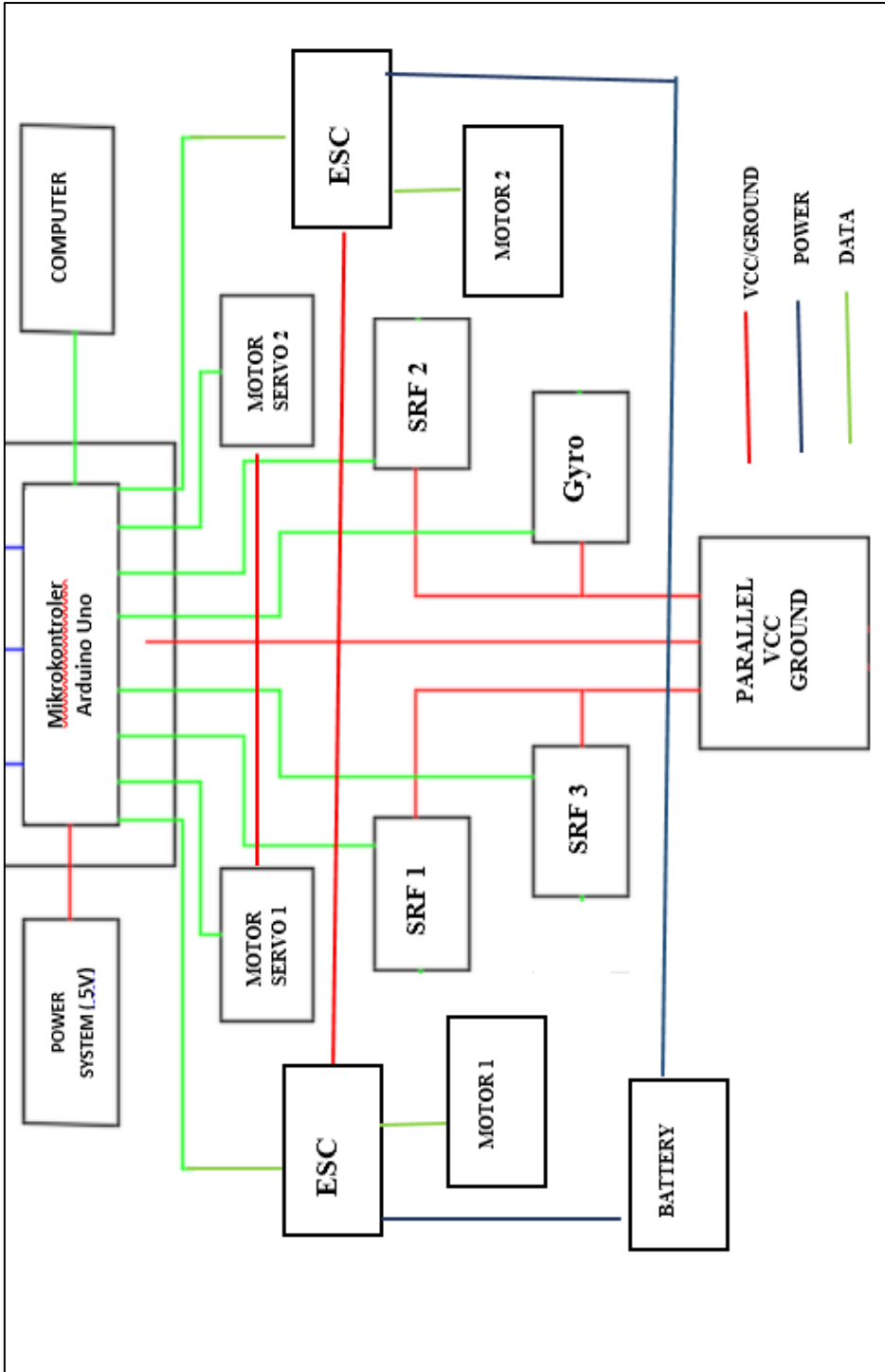
Tabel 4. 10 Spesifikasi Baterai

Komponen	Battery
Motor 1	4s ( 14.8 V)
Motor 2	3s ( 14.8 V)
Servo + SRF	Include in Arduino ( 5V)
Arduino	(5V) komputer

### 4.3.2 Lay out komponen Elektronika



Gambar 4. 6 Wiring Diagram



Gambar 4. 7 Skematik Elektronik

#### 4.5. Pemasangan Komponen

Berikut ini gambar komponen elektronik dan mekanik kapal yang sudah dirangkai :

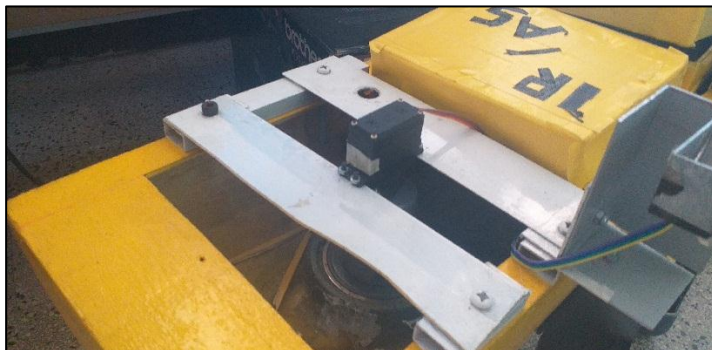
:



*Gambar 4. 8 Thruster Penggerak*

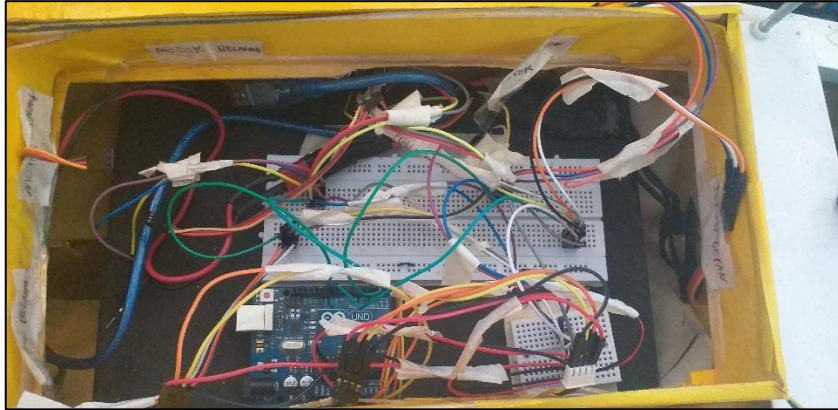


*Gambar 4. 9 Sensor Ultrasonik*



*Gambar 4. 10 Servo*





*Gambar 4. 11 Rangkaian Elektronik*



*Gambar 4. 12 Hasil Akhir Model*



#### 4. 6. Logika Pemograman

Logika Pemograman dibagi menjadi 4 menurut fungsi kerja masing masing

##### 4.5.1 Pemograman Sensor Ultrasonik

Berikut Kode Pemograman dan langkah langkah :

##### 1) Menentukan Variabel

```
// SRF Depan
```

```
const int TRI_depan = A4;
```

( SRF Depan keluaran Trigger ditentukan sebagai nilai interger dan variabel diberi nama variable “TRI\_depan” terhubung pada analog pin A4)

```
const int ECH_depan = A5;
```

( SRF Depan masukkan Echo ditentukan sebagai nilai interger dan variabel diberi nama variable “ECH\_depan” terhubung pada analog pin A5)

```
float duration_depan, distance_depan;
```

( Ditentukan sebagai nilai decimal variabel waktu tempuh gelombang dan diberi nama “duration\_depan”)

( Ditentukan sebagai nilai decimal variabel pembacaan jarak dan diberi nama “distance\_depan”)

```
// SRF Tengah
```

```
const int TRI_tengah = A2;
```

( SRF Tengah keluaran Trigger ditentukan sebagai nilai interger dan variabel diberi nama variable “TRI\_tengah” terhubung pada analog pin A2)

```
const int ECH_tengah = A3;
```

( SRF Tengah masukkan Echo ditentukan sebagai nilai interger dan variabel diberi nama variable “ECH\_tengah” terhubung pada analog pin A3)

**float duration\_tengah, distance\_tengah;**

( Ditentukan sebagai nilai decimal variabel waktu tempuh gelombang dan diberi nama “duration\_tengah”)

( Ditentukan sebagai nilai decimal variabel pembacaan jarak dan diberi nama “distance\_tengah”)

// SRF Belakang

**const int TRI\_belakang = A0;**

( SRF belakang keluaran Trigger ditentukan sebagai nilai interger dan variabel diberi nama variable “TRI\_belakang” terhubung pada analog pin A0)

**const int ECH\_belakang = A1;**

( SRF belakang masukkan Echo ditentukan sebagai nilai interger dan variabel diberi nama variable “ECH\_belakang” terhubung pada analog pin A1)

**float duration\_belakang, distance\_belakang;**

( Ditentukan sebagai nilai decimal variabel waktu tempuh gelombang dan diberi nama “duration\_belakang”)

( Ditentukan sebagai nilai decimal variabel pembacaan jarak dan diberi nama “distance\_belakang”)

**float distance\_total;**

( Ditentukan sebagai nilai decimal variabel pembacaan jarak dan diberi nama “distance\_total”)

## 2) Mendefinisikan Variabel

```
void setup() {  
    (Data atau Perintah yang akan dimasukan ke arduino)  
  
    // Variabel SRF  
    //SRF Depan  
    pinMode(TRI_depan, OUTPUT);  
    (SRF depan masukan Trigger didefinisikan sebagai output )  
    pinMode(ECH_depan, INPUT);  
    (SRF depan keluaran Echo didefinisikan sebagai input)  
  
    //SRF Tengah  
    pinMode(TRI_tengah, OUTPUT);  
    (SRF tengah masukan Trigger didefinisikan sebagai output )  
    pinMode(ECH_tengah, INPUT);  
    (SRF tengah keluaran Echo didefinisikan sebagai input)  
  
    //SRF Belakang  
    pinMode(TRI_belakang, OUTPUT);  
    (SRF belakang masukan Trigger didefinisikan sebagai output )  
    pinMode(ECH_belakang, INPUT);  
    (SRF tengah keluaran Echo didefinisikan sebagai input)
```

### 3) Mendapatkan Jarak

Gelombang ultrasonik merambat di udara dengan kecepatan 343 m/s pada suhu 20 derajat celcius. Waktu yang diukur merupakan waktu bolak balik gelombang, sehingga waktu rambat gelombang dibagi 2. Persamaan perhitungan menjadi :

$$\begin{aligned} D &= (DT / 2) \times C \\ &= (DT / 2) \times 0.0343 \\ &= (DT / 2) / 29.154 \end{aligned}$$

DT = Waktu Delay  
C = Kecepatan Rambat Suara  
D = Jarak

Kecepatan Suara = 343 m/s pada udara kering (20 \*C)  
= 0.0343 cm/mS

Waktu tempuh yang dibutuhkan untuk satu 1 cm = 29.154 mS

```
void loop() {
  (Data atau Perintah yang akan diolah arduino)
  // Pengolahan Data SRF
  // SRF Depan
  digitalWrite(TRI_depan, LOW);
  (memerintahkan sinyal digital kepada masukkan Trigger untuk tidak menyala)
  delayMicroseconds (2);
  (memerintahkan sinyal digital kepada masukkan Trigger untuk tidak menyala
  selama 2 mikrosekon)
  digitalWrite(TRI_depan, HIGH);
  (memerintahkan sinyal digital kepada masukkan Trigger untuk menyala)
  delayMicroseconds (10);
  (memerintahkan sinyal digital kepada masukkan Trigger untuk menyala selama 10
  mikrosekon)
```

**digitalWrite(TRI\_depan, LOW);**

(memerintahkan sinyal digital kepada masukkan Trigger untuk kembali tidak menyala)

**duration\_depan = pulseIn(ECH\_depan, HIGH);**

(membaca waktu yang ditempuh gelombang sampai masukkan Echo, saat masukkan Echo menyala)

**distance\_depan = (duration\_depan / 2) \* 0.0343;**

(menghitung jarak dalam cm dengan persamaan)

// SRF Tengah

**digitalWrite(TRI\_tengah, LOW);**

(memerintahkan sinyal digital kepada masukkan Trigger untuk tidak menyala)

**delayMicroseconds (2);**

(memerintahkan sinyal digital kepada masukkan Trigger untuk tidak menyala selama 2 mikrosekond)

**digitalWrite(TRI\_tengah, HIGH);**

(memerintahkan sinyal digital kepada masukkan Trigger untuk menyala)

**delayMicroseconds (10);**

(memerintahkan sinyal digital kepada masukkan Trigger untuk menyala selama 10 mikrosekond)

**digitalWrite(TRI\_tengah, LOW);**

(memerintahkan sinyal digital kepada masukkan Trigger untuk kembali tidak menyala)

**duration\_tengah = pulseIn(ECH\_tengah, HIGH);**

(membaca waktu yang ditempuh gelombang sampai masukkan Echo, saat masukkan Echo menyala)

**distance\_tengah = (duration\_tengah / 2) \* 0.0343;**

(menghitung jarak dalam cm dengan persamaan)

```
// SRF belakang
```

```
digitalWrite(TRI_belakang, LOW);
```

(memerintahkan sinyal digital kepada masukkan Trigger untuk tidak menyala)

```
delayMicroseconds (2);
```

(memerintahkan sinyal digital kepada masukkan Trigger untuk tidak menyala selama 2 mikrosekond)

```
digitalWrite(TRI_belakang , HIGH);
```

(memerintahkan sinyal digital kepada masukkan Trigger untuk menyala)

```
delayMicroseconds (10);
```

(memerintahkan sinyal digital kepada masukkan Trigger untuk menyala selama 10 mikrosekond)

```
digitalWrite(TRI_belakang, LOW);
```

(memerintahkan sinyal digital kepada masukkan Trigger untuk kembali tidak menyala)

```
duration_ = pulseIn(ECH_belakang, HIGH);
```

(membaca waktu yang ditempuh gelombang sampai masukkan Echo, saat masukkan Echo menyala)

```
distance_belakang = (duration_belakang /2) *0.0343;
```

(menghitung jarak dalam cm dengan persamaan)

```
distance_total = ((distance_depan + distance_tengah + distance_belakang) /3);
```

(jarak total rata rata juga dapat dihitung)

#### 4.5.2 Pemograman Gyro-Accelerometer

```

#include <Wire.h>
#include <Servo.h>

Servo right_prop;
Servo left_prop;
/*MPU-6050 gives you 16 bits data so you have to create some 16int
constants
* to store the data for accelerations and gyro*/

int16_t Acc_rawX, Acc_rawY, Acc_rawZ, Gyr_rawX, Gyr_rawY,
Gyr_rawZ;

float Acceleration_angle[2];
float Gyro_angle[2];
float Total_angle[2];

float elapsedTime, time, timePrev;
int i;
float rad_to_deg = 180/3.141592654;

float PID, pwmLeft, pwmRight, error, previous_error;
float pid_p=0;
float pid_i=0;
float pid_d=0;
//////////PID CONSTANTS//////////
double kp=3.55;//3.55
double ki=0.005;//0.003
double kd=2.05;//2.05

```

```
double throttle=1300; //initial value of throttle to the motors
float desired_angle = 0; //This is the angle in which we want the
                        //balance to stay steady

void setup() {
  Wire.begin(); //begin the wire communication
  Wire.beginTransmission(0x68);
  Wire.write(0x6B);
  Wire.write(0);
  Wire.endTransmission(true);
  Serial.begin(250000);
  right_prop.attach(3); //attach the right motor to pin 3
  left_prop.attach(5); //attach the left motor to pin 5

  time = millis(); //Start counting time in milliseconds
  /*In order to start up the ESCs we have to send a min value
  * of PWM to them before connecting the battery. Otherwise
  * the ESCs won't start up or enter in the configure mode.
  * The min value is 1000us and max is 2000us, REMEMBER!*/
  left_prop.writeMicroseconds(1000);
  right_prop.writeMicroseconds(1000);
  delay(7000); /*Give some delay, 7s, to have time to connect
               *the propellers and let everything start up*/
} //end of setup void

void loop() {
```



```

//////////////////////////////////I M U//////////////////////////////////
    timePrev = time; // the previous time is stored before the actual time read
    time = millis(); // actual time read
    elapsedTime = (time - timePrev) / 1000;

    /*The tiemStep is the time that elapsed since the previous loop.
    * This is the value that we will use in the formulas as "elapsedTime"
    * in seconds. We work in ms so we haveto divide the value by 1000
    to obtain seconds*/

    /*Reed the values that the accelerometre gives.
    * We know that the slave adress for this IMU is 0x68 in
    * hexadecimal. For that in the RequestFrom and the
    * begin functions we have to put this value.*/

    Wire.beginTransmission(0x68);
    Wire.write(0x3B); //Ask for the 0x3B register- correspond to AcX
    Wire.endTransmission(false);
    Wire.requestFrom(0x68,6,true);

    /*We have asked for the 0x3B register. The IMU will send a brust of
    register.
    * The amount of register to read is specify in the requestFrom function.
    * In this case we request 6 registers. Each value of acceleration is made
    out of
    * two 8bits registers, low values and high values. For that we request the 6
    of them
    * and just make then sum of each pair. For that we shift to the left the
    high values
    * register (<<) and make an or (|) operation to add the low values.*/

```

```

Acc_rawX=Wire.read()<<8|Wire.read(); //each value needs two registres
Acc_rawY=Wire.read()<<8|Wire.read();
Acc_rawZ=Wire.read()<<8|Wire.read();

//////This is the part where you need to calculate the angles using Euler
equations//////
    /* - Now, to obtain the values of acceleration in "g" units we first have to
divide the raw
    * values that we have just read by 16384.0 because that is the value that
the MPU6050
    * datasheet gives us.*/
    /* - Next we have to calculate the radian to degree value by dividing 180°
by the PI number
    * which is 3.141592654 and store this value in the rad_to_deg variable. In
order to not have
    * to calculate this value in each loop we have done that just once before
the setup void.
    */
    /* Now we can apply the Euler formula. The atan will calculate the
arctangent. The
    * pow(a,b) will elevate the a value to the b power. And finnaly sqrt
function
    * will calculate the rooth square.*/
    /*---X---*/
    Acceleration_angle[0] =
atan((Acc_rawY/16384.0)/sqrt(pow((Acc_rawX/16384.0),2) +
pow((Acc_rawZ/16384.0),2)))*rad_to_deg;
    /*---Y---*/
    Acceleration_angle[1] = atan(-
1*(Acc_rawX/16384.0)/sqrt(pow((Acc_rawY/16384.0),2) +
pow((Acc_rawZ/16384.0),2)))*rad_to_deg;

```

```

    /*Now we read the Gyro data in the same way as the Acc data. The
address for the
    * gyro data starts at 0x43. We can see this addresses if we look at the
register map
    * of the MPU6050. In this case we request just 4 values. W don;t want the
gyro for
    * the Z axis (YAW).*/
    Wire.beginTransmission(0x68);
    Wire.write(0x43); //Gyro data first address
    Wire.endTransmission(false);
    Wire.requestFrom(0x68,4,true); //Just 4 registers

    Gyr_rawX=Wire.read()<<8|Wire.read(); //Once again we shif and sum
    Gyr_rawY=Wire.read()<<8|Wire.read();

    /*Now in order to obtain the gyro data in degrees/seconda we have to
divide first
    the raw value by 131 because that's the value that the datasheet gives us*/
    /*---X---*/
    Gyro_angle[0] = Gyr_rawX/131.0;
    /*---Y---*/
    Gyro_angle[1] = Gyr_rawY/131.0;
    /*Now in order to obtain degrees we have to multiply the degree/seconds
    *value by the elapsedTime.*/
    /*Finnaly we can apply the final filter where we add the acceleration
    *part that affects the angles and ofcourse multiply by 0.98 */
    /*---X axis angle---*/
    Total_angle[0] = 0.98 *(Total_angle[0] + Gyro_angle[0]*elapsedTime) +
0.02*Acceleration_angle[0];
    /*---Y axis angle---*/
    Total_angle[1] = 0.98 *(Total_angle[1] + Gyro_angle[1]*elapsedTime) +
0.02*Acceleration_angle[1];

```



**/\*The last part is the derivate. The derivate acts upon the speed of the error. As we know the speed is the amount of error that produced in a certain amount of time divided by that time. For taht we will use a variable called previous\_error.**

**We substract that value from the actual error and divide all by the elapsed time.**

**Finnaly we multiply the result by the derivate constant\*/**

```
pid_d = kd*((error - previous_error)/elapsedTime);
```

**/\*The final PID values is the sum of each of this 3 parts\*/**

```
PID = pid_p + pid_i + pid_d;
```

**/\*We know taht the min value of PWM signal is 1000us and the max is 2000.**

**So that**

**tells us that the PID value can/s oscilate more than -1000 and 1000 because when we**

**have a value of 2000us the maximum value taht we could sybstract is 1000 and when**

**we have a value of 1000us for the PWM sihnal, the maximum value that we could add is 1000**

**to reach the maximum 2000us\*/**

```
if(PID < -1000)
```

```
{
```

```
    PID=-1000;
```

```
}
```

```
if(PID > 1000)
```

```
{
```

```
    PID=1000;
```

```
}
```

```
/*Finnaly we calculate the PWM width. We sum the desired throttle and the  
PID value*/
```

```
pwmLeft = throttle + PID;
```

```
pwmRight = throttle - PID;
```

```
/*Once again we map the PWM values to be sure that we won't pass the min  
and max values. Yes, we've already maped the PID values. But for example,  
for
```

```
throttle value of 1300, if we sum the max PID value we would have 2300us  
and
```

```
that will mess up the ESC.*/
```

```
//Right
```

```
if(pwmRight < 1000)
```

```
{
```

```
    pwmRight= 1000;
```

```
}
```

```
if(pwmRight > 2000)
```

```
{
```

```
    pwmRight=2000;
```

```
}
```

```
//Left
```

```
if(pwmLeft < 1000)
```

```
{
```

```
    pwmLeft= 1000;
```

```
}
```

```
if(pwmLeft > 2000)
```

```
{
```

```
    pwmLeft=2000;
```

```
}
```

```

/*Finnay using the servo function we create the PWM pulses with the
calculated
width for each pulse*/
left_prop.writeMicroseconds(pwmLeft);
right_prop.writeMicroseconds(pwmRight);
previous_error = error; //Remember to store the previous error.

} //end of loop void

```

#### 4.5.3 Pemograman Brushless DC Motor dan Servo

Berikut Kode pemograman dan langkah langkah

##### 1) Menentukan Variabel

```

#include <Servo.h>
(Mensertakan perintah output pengolahan sinyal )

// Variabel Motor
Servo Motor1;
(Thruster depan ditentukan sebagai motor dan diberi nama Motor1)
Servo Motor2;
(Thruster belakang ditentukan sebagai motor dan diberi nama Motor2)

//Variabel Servo

Servo Servo1;
(Servo depan ditentukan sebagai servo dan diberi nama Servo1)
Servo Servo2;
(Servo belakang ditentukan sebagai servo dan diberi nama Servo2)

```

## 2) Mendefinisikan Variabel

```
void setup() {
```

(Data atau Perintah yang akan dimasukan ke arduino)

```
//Variabel Motor
```

```
Motor1.attach(3);
```

( Motor1 didefinisikan sebagai output dan berada pada pin 9 )

```
Motor2.attach(5);
```

( Motor2 didefinisikan sebagai output dan berada pada pin 5 )

```
Servo1.attach(11);
```

( Servo1 didefinisikan sebagai output dan berada pada pin 11 )

```
Servo2.attach (10);
```

( Servo2 didefinisikan sebagai output dan berada pada pin 10 )

```
time = millis();
```

( waktu didefinisikan sebagai variabel dalam hitungan skala milisekon )

```
Motor1.writeMicroseconds(1000);
```

```
Motor2.writeMicroseconds(1000);
```

( Motor didefinisikan dalam keadaan mati atau (1000) pada nilai PWM saat controller belum memberikan perintah)

```
delay(1000);
```

( Motor dalam keadaan mati selama 1 detik sesaat sesudah menerima perintah controller)



### 3) Memasukkan perintah

```
void loop() {
```

(Data atau Perintah yang akan diolah arduino)

```
Motor1.writeMicroseconds(pwm1);
```

(memasukkan perintah sebesar nilai pwn hasil perhitungan PID kepada motor1)

```
Motor2.writeMicroseconds(pwm2);
```

(memasukkan perintah sebesar nilai pwn hasil perhitungan PID kepada motor2)

```
Servo1.write(180);
```

(memasukkan perintah sebesar nilai sudut yang diinginkan Servo1)

```
Servo2.write(180);
```

(memasukkan perintah sebesar nilai sudut yang diinginkan Servo2)

#### 4.5.4 Pemograman PID

Berikut Kode pemograman dan langkah langkah

##### 1) Menentukan Variabel

```
// Variabel PID
```

```
float elapsedTime, time, timePrev;
```

(ditentukan variable waktu pada sistem, yaitu elapsedTime, time, dan timePrev berupa bilangan decimal)

```
int i1;
```

```
int i2;
```

```
float PID1, pwm1 ,error1, previous_error1;
```

```
float PID2, pwm2 ,error2, previous_error2;
```

(ditentukan variable PID pada sistem, yaitu pwm, error, dan error sebelumnya berupa bilangan decimal)

```
float pid_p1=0;
```

```
float pid_i1=0;
```

```
float pid_d1=0;
```

```
float pid_p2=0;
```

```
float pid_i2=0;
```

```
float pid_d2=0;
```

```
double kp= 0;
```

```
double ki= 0;
```

```
double kd =0;
```

(Ditentukan nilai controller PID yang akan dimasukkan ke dalam sistem)

```
double throttle=1080;
```

(Ditentukan variabel dan nilai controller PID yang akan dimasukkan ke dalam sistem)

```
float desired_distance = 40;
```

(Ditentukan variable dan nilai controller jarak yang diinginkan sistem)

## 2) Perhitungan PID

```

void loop() {
// Perhitungan PID

timePrev = time;
(waktu sebelumnya merupakan waktu yang sudah berjalan)
time = millis();
(jumlah waktu sebenarnya sedang berjalan sejak program dijalankan)

elapsedTime = (time- timePrev)/1000;
(selisih waktu yang sudah berjalan sejak perintah terakhir dalam satuan second)

error1 = distance_depan-desired_distance;
error2 = distance_belakang-desired_distance;
(nilai error merupakan selisih jarak sebenarnya yang didapatkan atas pembacaan sensor dan jarak yang diinginkan)

pid_p1 = kp*error1;
pid_p2 = kp*error2;
(nilai kontrol proportional merupakan nilai propotional konstan dikali dengan nilai error)

if (-1000 <error1< 1000);
{
pid_i1 = pid_i1 +(ki*error1);
}

if (-1000 <error2< 1000);
{
pid_i2 = pid_i2 +(ki*error2);
}
(nilai kontrol integral merupakan nilai propotional integral dikali dengan nilai error kemudian ditambahkan dengan nilai integral itu sendiri. Hal ini akan mengintegral atau menambah nilai pada tiap loop)

```

```
pid_d1 = kd * ((error1 - previous_error1)/elapsedTime);
```

```
pid_d2 = kd * ((error2 - previous_error2)/elapsedTime);
```

(nilai kontrol derivative merupakan nilai propotional derivative dikali dengan nilai kecepatan perubahan error. Kecepatan perubahan error merupakan nilai error dikurangi nilai error sebelumnya pada tiap perubahan waktu)

```
PID1 = pid_p1 + pid_i1 + pid_d1;
```

```
PID2 = pid_p2 + pid_i2 + pid_d2;
```

(semua nilai controller ditambah menjadi nilai PID)

```
if(PID1 < -1000)
```

```
{
```

```
    PID1=-1000;
```

```
}
```

```
if(PID1 > 1000)
```

```
{
```

```
    PID1=1000;
```

```
}
```

```
if(PID2 < -1000)
```

```
{
```

```
    PID2=-1000;
```

```
}
```

```
if(PID2 > 1000)
```

```
{
```

```
    PID2=1000;
```

```
}
```

(nilai error kontrol PID dibatasi tidak lebih kurang dari nilai minimal PWM dan tidak melebihi dari nilai maksimal PWM )

```
pwm1 = throttle+PID1;
```

```
pwm2 = throttle+PID2;
```

(nilai PID ditambahkan dengan nilai throttle motor pada saat kondisi mempertahankan posisi sebagai nilai final PWM)

```
if(pwm1 < 1000)
```

```
{
```

```
  pwm1 = 1000;
```

```
}
```

```
if(pwm1 > 2000)
```

```
{
```

```
  pwm1 = 2000;
```

```
if(pwm2 < 1000)
```

```
{
```

```
  pwm2 = 1000;
```

```
}
```

```
if(pwm2 > 2000)
```

```
{
```

```
  pwm2 = 2000;
```

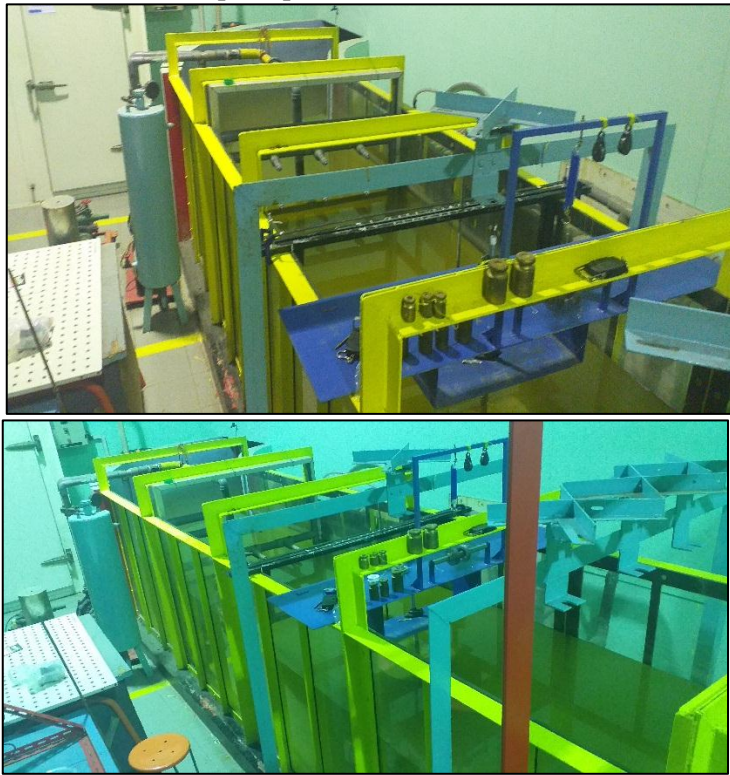
(nilai PWM total dibatasi tidak lebih kurang dari nilai minimal PWM dan tidak melebihi dari nilai maksimal PWM )

```
previous_error = error;
```

(nilai error sebelumnya disimpan untuk loop berikutnya )

#### 4. 7. Metode Pengujian

Pengujian dilakukan di kolam uji arus di Workshop Marine Machinery and System Departemen Teknik Sistem perkapalan.



*Gambar 4. 13 Kolam Uji*

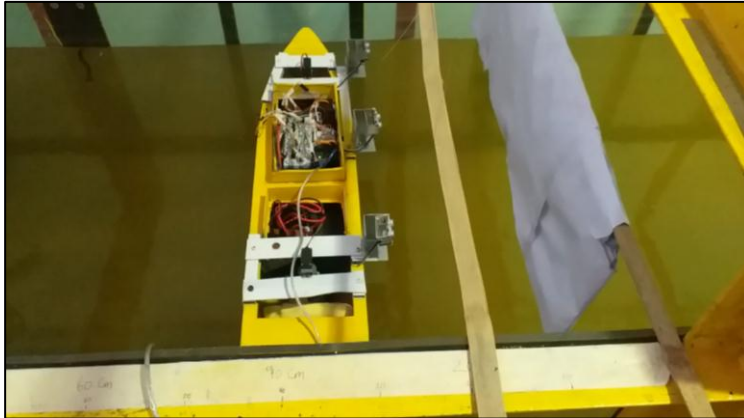
Sumber arus berasal dari air yang keluar dari nozzle yang digerakkan menggunakan pompa.



*Gambar 4. 14 Posisi Pengujian*

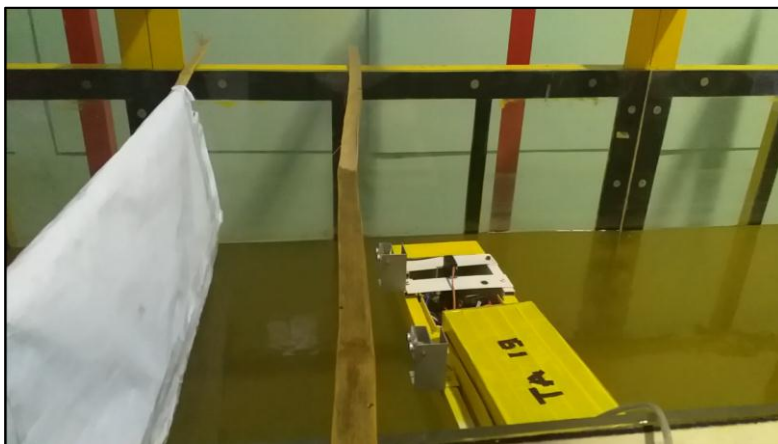
Digunakan papan sebagai referensi pembacaan dari sensor ultrasonic dan sebagai model dari oil rig di keadaan nyata.

Berikut gambar dari pengujian 1



*Gambar 4. 15 Pengujian 1*

Pada pengujian 1 arus datang dari sebelah Portside kapal. Dengan adanya arus maka kapal akan bergeser mendekati rig. Kuat arus yang keluar pada sumber berubah seiring dengan posisi pengujian dari sumber arus. Semakin jauh titik tertentu dari sumber arus maka arus akan semakin melemah. Berdasarkan pengukuran kecepatan arus pada titik keluar air dari sumber sebesar 0.010 m/s dan pada titik terjauh sebesar 0.001 m/s. Pengujian dilakukan pada tiap tiap jarak kapal dari rig yang ditentukan. Dengan nilai estimasi jarak, maka kapal harus berada pada titik tertentu sebesar jarak tertentu dari rig. Jarak itu sebesar 20 cm, 40 cm, 60 cm, 80 cm, 100 cm. Pengujian dilakukan dari kondisi error maksimal, yaitu saat kapal berada jauh dari titik referensi yaitu saat kapal berada sedekat mungkin di depan rig.



*Gambar 4. 16 Pengujian 2*

Pada pengujian 2 arus datang dari sebelah Portside kapal. Dengan adanya arus maka kapal akan bergeser menjauhi model rig. Kuat arus yang keluar pada sumber berubah seiring dengan posisi pengujian dari sumber arus. Semakin jauh titik tertentu dari sumber arus maka arus akan semakin melemah. Berdasarkan pengukuran kecepatan arus pada titik keluar air dari sumber sebesar 0.010 m/s dan pada titik terjauh sebesar 0.001 m/s. Pengujian dilakukan pada tiap tiap jarak kapal dari rig yang ditentukan.

Dengan nilai estimasi jarak, maka kapal harus berada pada titik tertentu sebesar jarak tertentu dari rig. Jarak itu sebesar 20 cm, 40 cm, 60 cm, 80 cm, 100 cm. Pengujian dilakukan dari kondisi error maksimal, yaitu saat kapal berada jauh dari titik referensi yaitu saat kapal berada sejauh mungkin dari rig.

Dengan sistem kontrol maka kapal diperintahkan untuk bergerak melawan kecepatan arus, serta memperkecil nilai error. Nilai error menjadi 0 saat kapal berada pada titik yang diperintahkan. Nilai error merupakan jarak yang ditentukan dikurangi jarak sebenarnya yang terukur oleh sensor. Respon akhir kapal terhadap jarak yang ditentukan mungkin dapat melebihi atau kurang dari jarak yang ditentukan tersebut. Nilai tersebut akan menjadi nilai simpangan dari respon program.

Sebelum pengujian final akan dilakukan trial dan error terlebih dahulu. Trial dan error berguna untuk mendapatkan nilai tuning PID dan nilai kecepatan minimal motor untuk mempertahankan posisi diam akibat gaya dorong dari arus arus. Tuning PID dilakukan sebelum pengujian final dengan metode tuning trial dan error. Tuning dilakukan dengan memasukkan nilai tertentu berdasarkan pengamatan dari pengujian. Nilai tuning yang dirasa cukup akan menjadi patokan tuning tetap pada tiap kondisi pengujian akhir. Hasil pengujian akhir merupakan nilai pendekatan dan observasi selama pengujian. Variabel yg ada pada percobaan meliputi

- **Jarak yang ditentukan**  
Jarak kapal yang ditentukan dari rig pada titik tertentu. Kapal harus mencapai titik tersebut untuk mengurangi nilai error.
- **Kecepatan Arus Air**  
Kecepatan arus air yang pada titik pengamatan. Kecepatan arus maksimal terdapat pada titik keluarnya air dari sumber. Arus semakin melemah pada titik yang semakin menjauhi sumber arus. Pada kondisi kenyataan saat pengujian, aliran air bersifat turbulen dan tidak laminar. Kecepatan Arus yang mengenai badan kapal tidak sama pada tiap titik. Untuk mempermudah pengukuran, diasumsikan bahwa kecepatan arus hanya dibagi menjadi 2 titik, yaitu arus yang mengenai titik bagian haluan kapal dan arus yang mengenai titik buritan kapal. Respon putaran thruster pada kapal berbeda satu sama lain, bergantung dari nilai error yang dibaca oleh sensor preferensi. Sebagai contoh : Thruster bagian haluan menerima perintah dari kontroler berdasarkan pembacaan sensor preferensi di bagian haluan, begitu juga dengan bagian buritan kapal.
- **Jarak Simpangan error**  
Jarak simpangan error merupakan jarak simpangan/osilasi kapal saat kapal sudah berada dalam posisi statis.
- **Nilai kecepatan Motor Saat kondisi stabil**  
Merupakan nilai putaran motor untuk mempertahankan posisi pada titik tertentu dengan gaya tertentu yang diberikan oleh arus air.
- **Nilai Tuning PID**  
Merupakan nilai tuning konstanta  $K_p$ ,  $K_i$ ,  $K_d$ . Nilai ini didapat dengan metode trial and error. Nilai yang menghasilkan respon yang baik akan dijadikan nilai tuning pada setiap keadaan percobaan.
- **Respon Waktu**  
Merupakan waktu yang dibutuhkan untuk mencapai jarak yang ditentukan.



#### 4. 8. Tuning PID

Sebelum melakukan pengujian final, dilakukan tuning nilai PID berdasarkan metode Zieger Nichols. Tuning PID dilakukan dengan metode :

- Meruning program Arduino dengan tuning PID masing masing 0 pada satu kecepatan gelombang tertentu
- Menambah nilai tuning Propotional P secara bertahap sampai system membentuk kurva step plant berbentuk S
- Kurva diamati melalui serial plotter pada Arduino



Gambar 4. 17 Kurva Arduino

- Didapati nilai T dan L yaitu 7 dan 3

Tipe kontroler	$K_p$	$\tau_i$	$\tau_d$
P	$\frac{T}{L}$	$\infty$	0
PI	$0,9\frac{T}{L}$	$\frac{L}{0,3}$	0
PID	$1,2\frac{T}{L}$	2L	0,5L

- Sehingga nilai PID didapati yaitu :  
 $K_p = 0.5 * 7/2 = 1.2$  ,  
 $K_i = 1/ (2*3) = 1/6 = 0.2$   
 $K_d = 0.5* 7 = 3.5$   
 Nilai dimasukkan pada program arduino

#### 4. 9. Mencari nilai PWM minimal

Nilai PWM minimal merupakan nilai yang akan dikonversikan menjadi nilai minimal kecepatan tiap percobaan. Nilai putaran motor minimum merupakan nilai putaran motor untuk mempertahankan posisi pada titik tertentu dengan gaya tertentu yang diberikan oleh arus air. Hasil didapatkandengan memasukkan nilai kecepatan secara kepada program secara manual tanpa PID dan menyesuaikan dengan jarak hasil pengamatan. Posisi diam merupakan posisi kecepatan minimal berada dan dicatat. Berikut didapatkan nilai PWM motor pada masing masing pengujian :

- Pengujian 1

*Tabel 4. 11 Hasil Trial And Error*

<b>Jarak Yang Ditentukan (Cm)</b>	<b>PWM</b>	<b>RPM</b>	<b>Thrust (Kg)</b>
20	1069	358	0.095
40	1074	384	0.102
60	1076	395	0.105
80	1078	405	0.108
100	1080	416	0.111

- Pengujian 2  
Pada pengujian 2 nilai dibalik karena posisi platform berpindah di sebelah kiri kapal

*Tabel 4. 12 Tabel PWM*

<b>Jarak Yang Ditentukan (Cm)</b>	<b>PWM</b>	<b>RPM</b>	<b>KG</b>
100	1069	358	0.095
80	1074	384	0.102
60	1076	395	0.105
40	1078	405	0.108
20	1080	416	0.111

#### 4. 10. Hasil Pengujian

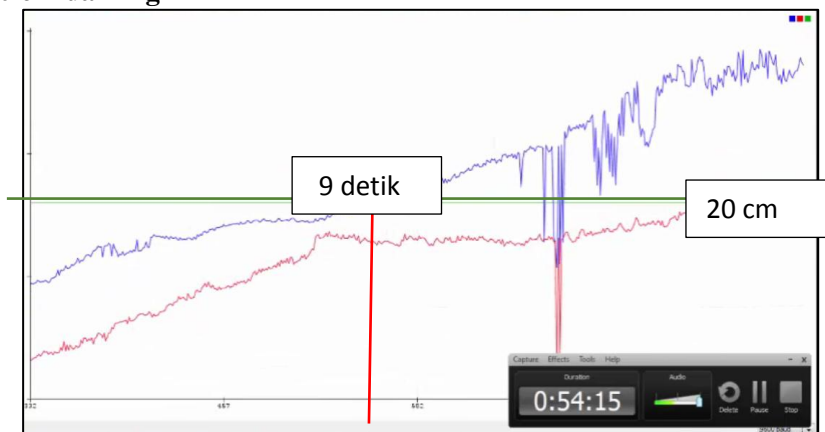
Setelah didapatkan nilai tuning PID dan nilai PWM minimal, barulah nilai tersebut dimasukkan ke dalam program. Kemudian dilakukan pengujian final pada kapal. Variabel yang ditinjau sesuai dengan yang dijelaskan di bagian sebelumnya. Berikut hasil pengujian

##### 1) Pengujian Gerakan 1

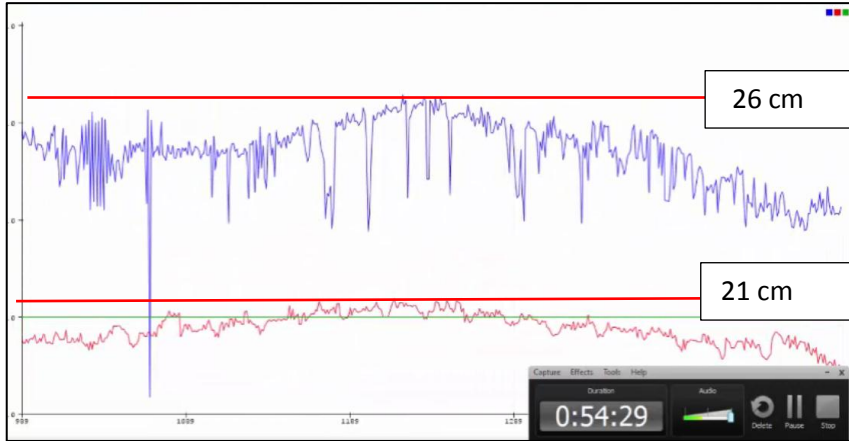
Tabel 4. 13 Hasil Pengujian 1

Jarak Yang Ditetapkan (Cm)	Nilai simpangan jarak/ error Bagian haluan dan buritan (cm)	Respon Waktu mencapai set point	Nilai tuning PID
20	6	9 s	P=1.2 I=0.2 D=3.5
	1		
40	7	8 s	P=1.2 I=0.2 D=3.5
	3		
60	7	9 s	P=1.2 I=0.2 D=3.5
	4		
80	5	9 s	P=1.2 I=0.2 D=3.5
	2		
100	5	10 s	P=1.2 I=0.2 D=3.5
	3		

##### Jarak 20 cm dari rig

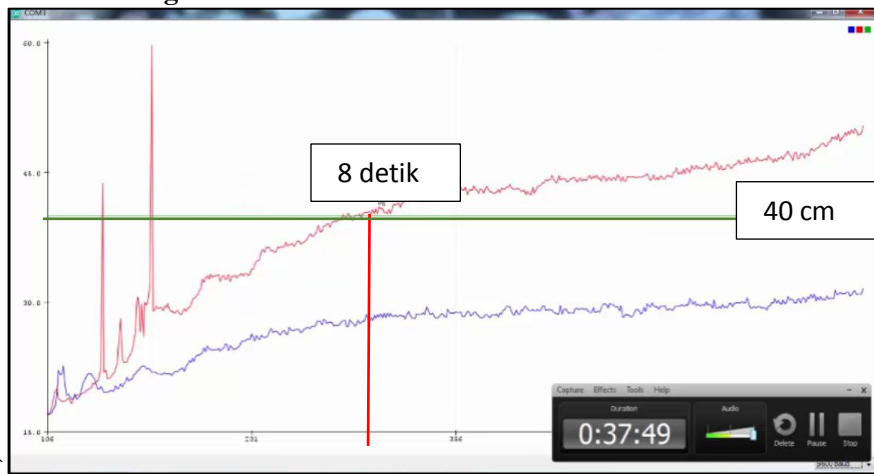


Gambar 4. 18 Kurva Arduino 20 cm

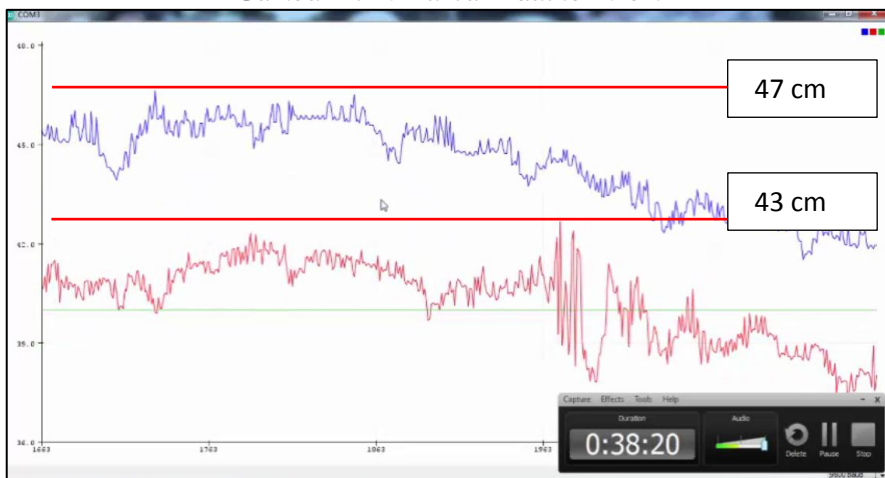


Gambar 4. 19 Kurva Arduino 20 cm

**Jarak 40 cm dari rig**

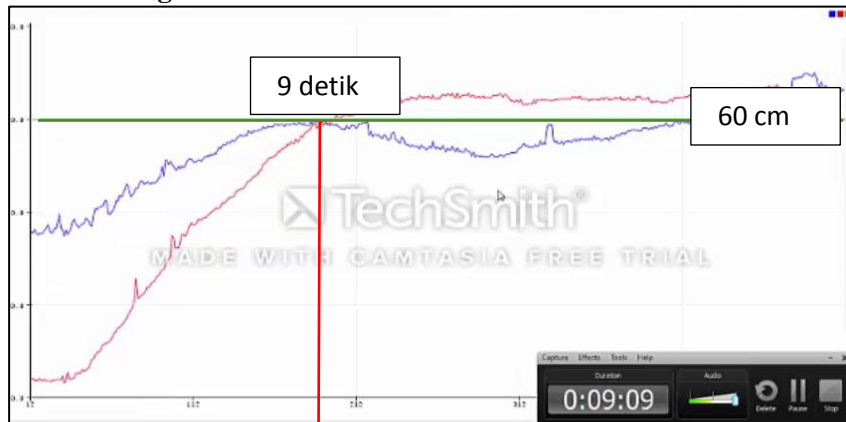


Gambar 4. 20 Kurva Arduino 40 cm



Gambar 4. 21 Kurva Arduino 40 cm

### Jarak 60 cm dari rig



Gambar 4. 22 Kurva Arduino 60 cm

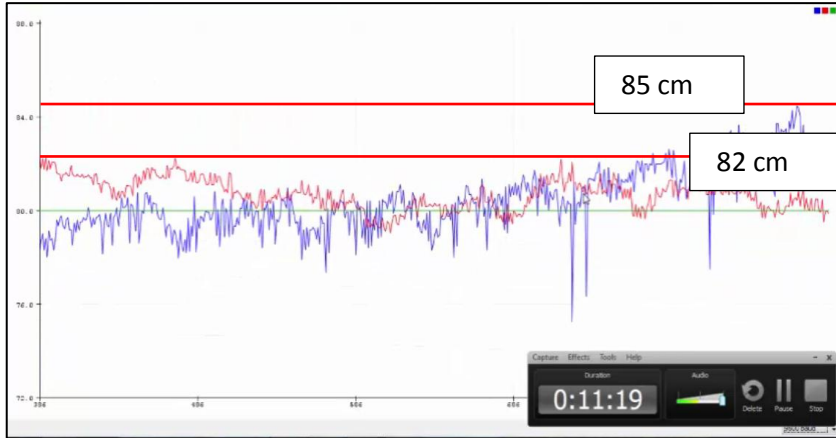


Gambar 4. 23 Kurva Arduino 60 cm

### Jarak 80 cm dari rig

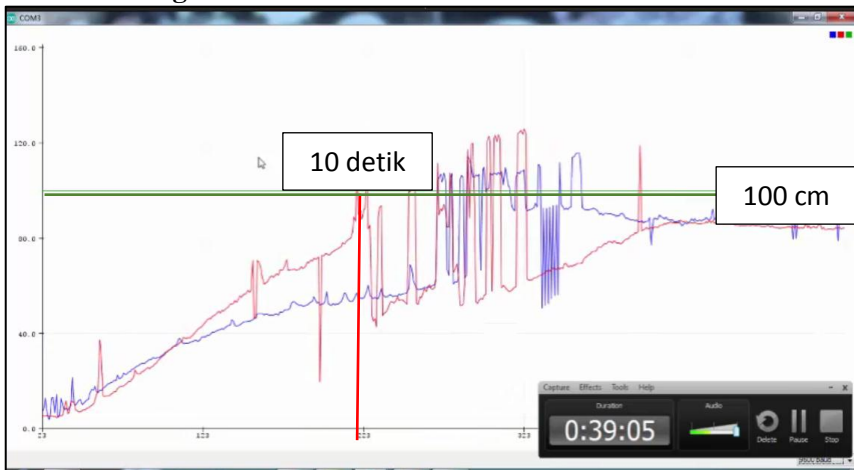


Gambar 4. 24 Kurva Arduino 80 cm

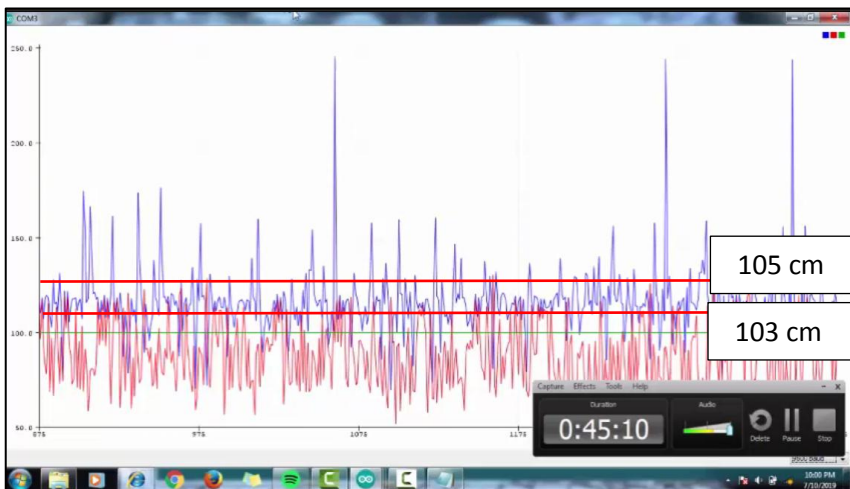


Gambar 4. 25 Kurva Arduino 80 cm

Jarak 100 cm dari rig



Gambar 4. 26 Kurva Arduino 100 cm



Gambar 4. 27 Kurva Arduino 100 cm

## 2) Pengujian Gerakan 2

*Tabel 4. 14 Hasil Pengujian 2*

Jarak Yang Ditetapkan (Cm)	Nilai simpangan jarak/ error Bagian haluan dan buritan (cm)	Respon Waktu mencapai set point	Nilai tuning PID
20	5	10 s	P=1.2 I=0.2 D=3.5
	2		
40	7	9 s	P=1.2 I=0.2 D=3.5
	3		
60	6	9 s	P=1.2 I=0.2 D=3.5
	5		
80	6	8s	P=1.2 I=0.2 D=3.5
	4		
100	6	9 s	P=1.2 I=0.2 D=3.5
	3		

### 4. 11. Pembahasan

Berdasarkan hasil pengujian didapati bahwa kapal memiliki nilai putaran motor berbeda beda. Pada pengujian 1 nilai putaran terbesar sebesar 416 rpm pada jarak 100 cm , yang mana merupakan jarak terjauh dari model rig. Sedangkan pada percobaan 2 putaran motor terbesar, 416 rpm berada pada jarak terdekat dari model rig yaitu 20 cm. Dari data tersebut dapat disimpulkan bawa putaran motor pada kapal bergantung pada besarnya gaya luar yang diterima kapal akibat kecepatan arus air. Semakin dekat jarak pengujian dengan sumber arus air, maka diperlukan putaran motor yang lebih besar untuk mempertahankan posisi diam dari kapal di titik tersebut.

Berdasarkan pengujian 1 dan 2, nilai rata rata simpangan jarak (error) sebesar 0.06 m pada bagian haluan . Berdasarkan pengujian 1 nilai rata rata simpangan jarak (error) sebesar 0.026 m pada bagian buritan sedangkan pada pengujian 2 sebesar 0.034 m diukur dari set point. Dari data tersebut dapat disimpulkan bahwa selama pengujian bagian haluan kapal selalu berada lebih di depan dari bagian buritan kapal diukur dari set point. Hal ini dapat terjadi kemungkinan dikarenakan gaya luar yang diterima kapal pada bagian haluan lebih kecil dibandingkan dengan bagian buritan. Bagian haluan kapal memiliki luas permukaan lebih kecil dibanding bagian buritan kapal. Sehingga dengan gaya putaran motor yang sama, maka bagian haluan kapal akan lebih mudah untuk mengeliminir gaya luar dari arus air.

Respon waktu rata rata mencapai set point bervariasi dengan nilai rata rata sebesar 9 detik . Dari hasil pengujian dapat disimpulkan bahwa respon waktu setiap percobaan sama, yaitu bergantung dengan konstanta PID yang digunakan.

“ Halaman ini sengaja dikosongkan”



## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5. 1. Kesimpulan**

Berdasarkan penelitian yang sudah dilakukan dalam tugas akhir ini didapatkan kesimpulan bahwa :

- 1) Rancang Bangun Dynamic Positioning System pada model model Supply Vessel menggunakan Sensor ultrasonik dengan kontrol Arduino Berbasis PID dapat berguna untuk mendapatkan gambaran dan sebagai pembanding dari System Dynamic Positioning di keadaan sebenarnya.
- 2) Hasil yang didapatkan dari pengujian adalah kapal mampu merepresentasikan respon untuk mempertahankan posisinya pada titik dan arah tertentu dengan nilai simpangan jarak ( error) berdasarkan pengujian 1 dan 2, sebesar 0.06 m pada bagian haluan, berdasarkan pengujian 1 sebesar 0.026 m pada bagian buritan dan pada pengujian 2 sebesar 0.034 m pada bagian buritan diukur dari set point.
- 3) Berdasarkan hasil pengujian didapati bahwa kapal memiliki nilai putaran motor berbeda beda. Pada pengujian 1 nilai putaran terbesar sebesar 416 rpm pada jarak setpoint 100 cm , yang mana merupakan jarak terjauh dari model rig. Sedangkan pada percobaan 2 putaran motor terbesar, 416 rpm berada pada jarak set point terdekat dari model rig yaitu 20 cm. Dari data tersebut dapat disimpulkan bawa putaran motor pada kapal bergantung pada besarnya gaya luar yang diterima kapal akibat kecepatan arus air. Semakin dekat jarak pengujian dengan sumber arus air, maka diperlukan putaran motor yang lebih besar untuk mempertahankan posisi diam dari kapal di titik tersebut.
- 4) Respon waktu rata rata mencapai set point berdasarkan pengujian 1 dan 2 bervariasi dengan nilai rata rata sebesar 9 detik . Dari hasil pengujian didapati bahwa respon waktu set point setiap percobaan memiliki nilai yang berdekatan dengan waktu terkecil 8 detik pada percobaan 1 setpoint 40 cm dan waktu terbesar 10 detik pada percobaan 2 setpoint 20 cm. Jadi dapat disimpulkan bahwa respon waktu setiap percobaan memiliki nilai yang berdekatan, yaitu bergantung dengan konstanta PID yang digunakan.
- 5) Kesalahan dan nilai error pada sistem dapat terjadi karena berbagai factor. Diantaranya factor internal yaitu kesalahan pembacaan data akibat akurasi sensor. Faktor eksternal lingkungan uji yaitu bentuk lambung kapal dan aliran air yang tidak konstan dan turbulen.

#### **5. 2. Saran**

Berikut merupakan saran dari penulis terhadap pembuatan dan pengujian model dalam pengerjaan tugas akhir :

- 1) Dilakukan perhitungan pemodelan matematis dari desain serta respon pergerakan agar didapatkan persamaan yang akurat dan dapat dibandingkan dengan data percobaan
- 2) Pada penelitian selanjutnya untuk pengembangan secara teknis, hal yang dapat dilakukan diantaranya yaitu : menggunakan jenis sensor referensi lain yang memiliki tingkat keakuratan yang lebih tinggi seperti sensor cahaya Lidar, dan tambahan sensor lain seperti angin dan vertical reference.

“ Halaman ini sengaja dikosongkan”

## DAFTAR PUSTAKA

- T. I. Fossen, *Guidance and Control of Ocean Vehicles*, John Wiley and Sons Ltd, 1994.
- A. A. Aly, "PID Parameters Optimization Using Genetic Algorithm Technique for Electrohydraulic Servo Control System," *Intelligent Control and Automation*, no. 2, pp. 69-76, 2011.
- ABS (American Bureau of Shipping), 2013. Classification Notation. Dalam: *Guide for Dynamic Positioning System*. s.l.:s.n., p.1.
- Chas, C. S. & Ferreiro, R., 2008. *Introduction To Ship Dynamic Positioning System*.
- Morgan, M. J., 1978. PPC Books Division. *Dynamic Positioning of Offshore Vessels*.*Journal of Maritime Research*.
- Research Journal of Science & IT Management*, vol. 4, 2015.
- The Nautical Institute, 2011
- International Marine Contractors Association, 2003
- The Basic Principles and Use of Hydroacoustic Position Reference Systems in the Offshore Environment*, 1999
- The Basic Principles and Use of Hydroacoustic Position Reference Systems in the Offshore Environment*, 1999
- A Review of Marine Laser Positioning Systems*, 2003
- Nurlita Gamayanti, *DESAIN KONTROLER PID ZIEGLER NICHOLS*  
[www.maritimeworld.web.id](http://www.maritimeworld.web.id)  
Arduino.CC  
[howtomechatronics.com](http://howtomechatronics.com)

“ Halaman ini sengaja dikosongkan”

**LAMPIRAN A**  
**PROGRAM ARDUINO POSITION REFERENCE**

/\*Programming Code Arduino TES 7

TUGAS AKHIR ALFIUS

RANCANG BANGUN DYNAMIC POSITIONING SYSTEM PADA MODEL  
MODEL SUPPLY VESSEL MENGGUNAKN SENSOR SRF DENGAN KONTROL  
ARDUINO BERBASIS PID

Alfius Christian Katilik  
NRP. 04211541000003

Dosen Pembimbing  
Indra Ranu Kusuma, S.T., M.Sc.  
Ir. Agoes Santoso, M.Sc., M.phil.

\*/

```
#include <Servo.h>
```

```
// Variabel PID
```

```
float elapsedTime, time, timePrev;  
int i1;  
int i2;
```

```
float PID1, pwm1 ,error1, previous_error1;  
float PID2, pwm2 ,error2, previous_error2;
```

```
float pid_p1=0;  
float pid_i1=0;  
float pid_d1=0;
```

```
float pid_p2=0;  
float pid_i2=0;  
float pid_d2=0;
```

```
// Menentukan nilai PID

double kp= 2;
double ki= 0.003;
double kd =1.2;

double throttle=1080;
float desired_distance = 40;

// Variabel Motor

Servo Motor1;
Servo Motor2;

// Variabel SRF

// SRF Depan
const int TRI_depan = A4; // SRF depan keluaran Trigger ada di analog pin A4
const int ECH_depan = A5; // SRF depan masukan Echo ada di analog pin A5
float duration_depan, distance_depan;

// SRF Tengah
const int TRI_tengah = A2; // SRF tengah keluaran Trigger ada di analog pin A2
const int ECH_tengah = A3; // SRF tengah masukan Echo ada di analog pin A3
float duration_tengah, distance_tengah;

// SRF Belakang
const int TRI_belakang = A0; // SRF tengah keluaran Trigger ada di analog pin A0
const int ECH_belakang = A1; // SRF tengah masukan Echo ada di analog pin A1

float duration_belakang, distance_belakang;

float distance_total;

void setup() {
  // Data atau Perintah yang akan dimasukkan ke arduino

  Serial.begin(9600);
```

```

// Variabel SRF
//SRF Depan
pinMode(TRI_depan, OUTPUT); // SRF depan masukan Trigger sebagai output
pinMode(ECH_depan, INPUT); // SRF depan keluaran Echo sebagai input

//SRF Tengah
pinMode(TRI_tengah, OUTPUT); // SRF tengah masukan Trigger sebagai output
pinMode(ECH_tengah, INPUT); // SRF tengah keluaran Echo sebagai input

//SRF Belakang
pinMode(TRI_belakang, OUTPUT); // SRF belakang masukan Trigger sebagai output
pinMode(ECH_belakang, INPUT); // SRF belakang keluaran Echo sebagai input

//Variabel Motor

  Motor1.attach(3); // Motor depan pada pin 3 sebagai output berlaku perintah sebagai
output
  Motor2.attach(5); // Motor depan pada pin 5 sebagai output berlaku perintah sebagai
output

  time = millis();

  Motor1.writeMicroseconds(1000);
  Motor2.writeMicroseconds(1000);

  delay(1000);

}

void loop() {

// MonitorData

  Serial.print("distance_depan=");
  Serial.print(distance_depan);
  Serial.println("cm");
  Serial.print("distance_tengah=");
  Serial.print(distance_tengah);
  Serial.println("cm");
  Serial.print("distance_belakang=");
  Serial.print(distance_belakang);
  Serial.println("cm");

  Serial.print(pwm1);
  Serial.print (" ");
  Serial.println(pwm2);

```

```
Serial.print(error1);
Serial.print (" ");
Serial.println(error2);
```

```
// Pengolahan Data SRF
// SRF Depan
```

```
digitalWrite(TRI_depan, LOW);
delayMicroseconds (2); // membuat awalan trigger pin selama sementara waktu
berhenti
```

```
digitalWrite(TRI_depan, HIGH);
delayMicroseconds (10); // menyalakan trigger pin selama 10 microseconds
digitalWrite(TRI_depan, LOW);
```

```
duration_depan = pulseIn(ECH_depan, HIGH); //membaca waktu dari perintah
sebelumnya gelombang echo pin
distance_depan = (duration_depan /2) *0.0343; // menghitung jarak dalam cm
```

```
// SRF Tengah
```

```
digitalWrite(TRI_tengah, LOW);
delayMicroseconds (2); // membuat awalan trigger pin selama sementara waktu
berhenti
```

```
digitalWrite(TRI_tengah, HIGH);
delayMicroseconds (10); // menyalakan trigger pin selama 10 microseconds
digitalWrite(TRI_tengah, LOW);
```

```
duration_tengah = pulseIn(ECH_tengah, HIGH); //membaca waktu dari perintah
sebelumnya gelombang echo pin
distance_tengah = (duration_tengah /2) *0.0343 ; // menghitung jarak dalam cm
```

```
// SRF Belakang
```

```
digitalWrite(TRI_belakang, LOW);
delayMicroseconds (2); // membuat awalan trigger pin selama sementara waktu
berhenti
```

```
digitalWrite(TRI_belakang, HIGH);
delayMicroseconds (10); // menyalakan trigger pin selama 10 microseconds
```



```

digitalWrite(TRI_belakang, LOW);

duration_belakang = pulseIn(ECH_belakang, HIGH); //membaca waktu dari perintah
sebelumnya gelombang echo pin

distance_belakang = (duration_belakang /2) *0.0343; // menghitung jarak dalam cm

distance_total = ((distance_depan + distance_tengah + distance_belakang) /3);

// Perhitungan PID

timePrev = time;
time = millis();

elapsedTime = (time- timePrev)/1000;

error1 = desired_distance-distance_depan;
error2 = desired_distance-distance_belakang;

pid_p1 = kp*error1;
pid_p2 = kp*error2;

if (-10 <error1< 10);
{
pid_i1 = pid_i1 +(ki*error1);
}
pid_d1 = kd * ((error1 - previous_error1)/elapsedTime);
PID1 = pid_p1 + pid_i1 + pid_d1;

if (-10 <error2< 10);
{
pid_i2 = pid_i2 +(ki*error2);
}
pid_d2 = kd * ((error2 - previous_error2)/elapsedTime);

PID2 = pid_p2 + pid_i2 + pid_d2;

```

```
if(PID1 < -1000)
{
  PID1=-1000;
}
if(PID1 > 1000)
{
  PID1=1000;
}
```

```
if(PID2 < -1000)
{
  PID2=-1000;
}
if(PID2 > 1000)
{
  PID2=1000;
}
```

```
pwm1 = 0;
pwm2 = 0;
```

```
if(pwm1 < 1000)
{
  pwm1 = 1000;
}
if(pwm1 > 2000)
{
  pwm1 = 2000;
}
```

```
if(pwm2 < 1000)
{
  pwm2 = 1000;
}
if(pwm2 > 2000)
{
  pwm2 = 2000;
}
```

```
Motor1.writeMicroseconds(pwm1); // memasukkan perintah besar nilai pwn hasil
perhitungan PID
Motor2.writeMicroseconds(pwm2); // hasil yg sama dimasukkan ke dalam motor 2
previous_error1 = error1;
previous_error2 = error2;
}
```

## LAMPIRAN B

### PROGRAM ARDUINO HEADING REFERENCE

```

#include <Wire.h>
#include <Servo.h>

Servo right_prop;
Servo left_prop;

/*MPU-6050 gives you 16 bits data so you have to create some 16int constants
 * to store the data for accelerations and gyro*/

int16_t Acc_rawX, Acc_rawY, Acc_rawZ,Gyr_rawX, Gyr_rawY, Gyr_rawZ;

float Acceleration_angle[2];
float Gyro_angle[2];
float Total_angle[2];

float elapsedTime, time, timePrev;
int i;
float rad_to_deg = 180/3.141592654;

float PID, pwmLeft, pwmRight, error, previous_error;
float pid_p=0;
float pid_i=0;
float pid_d=0;

//////////PID CONSTANTS//////////
double kp=3.55;//3.55
double ki=0.005;//0.003
double kd=2.05;//2.05
//////////

double throttle=1300; //initial value of throttle to the motors
float desired_angle = 0; //This is the angle in which we want the
//balance to stay steady

```

```

void setup() {
  Wire.begin(); //begin the wire communication
  Wire.beginTransmission(0x68);
  Wire.write(0x6B);
  Wire.write(0);
  Wire.endTransmission(true);
  Serial.begin(250000);
  right_prop.attach(3); //attach the right motor to pin 3
  left_prop.attach(5); //attach the left motor to pin 5

  time = millis(); //Start counting time in milliseconds
  /*In order to start up the ESCs we have to send a min value
  * of PWM to them before connecting the battery. Otherwise
  * the ESCs won't start up or enter in the configure mode.
  * The min value is 1000us and max is 2000us, REMEMBER!*/
  left_prop.writeMicroseconds(1000);
  right_prop.writeMicroseconds(1000);
  delay(7000); /*Give some delay, 7s, to have time to connect
  *the propellers and let everything start up*/
} //end of setup void

void loop() {

  ////////////////////////////////////I M U////////////////////////////////////
  timePrev = time; // the previous time is stored before the actual time read
  time = millis(); // actual time read
  elapsedTime = (time - timePrev) / 1000;

  /*The tiemStep is the time that elapsed since the previous loop.
  * This is the value that we will use in the formulas as "elapsedTime"
  * in seconds. We work in ms so we haveto divide the value by 1000
  to obtain seconds*/

  /*Reed the values that the accelerometer gives.
  * We know that the slave adress for this IMU is 0x68 in
  * hexadecimal. For that in the RequestFrom and the
  * begin functions we have to put this value.*/

  Wire.beginTransmission(0x68);
  Wire.write(0x3B); //Ask for the 0x3B register- correspond to AcX
  Wire.endTransmission(false);
  Wire.requestFrom(0x68,6,true);

  /*We have asked for the 0x3B register. The IMU will send a brust of register.

```

```

* The amount of register to read is specify in the requestFrom function.
* In this case we request 6 registers. Each value of acceleration is made out of
* two 8bits registers, low values and high values. For that we request the 6 of them
* and just make then sum of each pair. For that we shift to the left the high values
* register (<<) and make an or (|) operation to add the low values.*/

```

```

Acc_rawX=Wire.read()<<8|Wire.read(); //each value needs two registres
Acc_rawY=Wire.read()<<8|Wire.read();
Acc_rawZ=Wire.read()<<8|Wire.read();

```

```

/*///This is the part where you need to calculate the angles using Euler equations///*/

```

```

/* - Now, to obtain the values of acceleration in "g" units we first have to divide the
raw

```

```

* values that we have just read by 16384.0 because that is the value that the
MPU6050

```

```

* datasheet gives us.*/

```

```

/* - Next we have to calculate the radian to degree value by dividing 180° by the PI
number

```

```

* which is 3.141592654 and store this value in the rad_to_deg variable. In order to
not have

```

```

* to calculate this value in each loop we have done that just once before the setup
void.

```

```

*/

```

```

/* Now we can apply the Euler formula. The atan will calculate the arctangent. The

```

```

* pow(a,b) will elevate the a value to the b power. And finnaly sqrt function

```

```

* will calculate the rooth square.*/

```

```

/*---X---*/

```

```

Acceleration_angle[0] =
atan((Acc_rawY/16384.0)/sqrt(pow((Acc_rawX/16384.0),2) +
pow((Acc_rawZ/16384.0),2)))*rad_to_deg;

```

```

/*---Y---*/

```

```

Acceleration_angle[1] = atan(-
1*(Acc_rawX/16384.0)/sqrt(pow((Acc_rawY/16384.0),2) +
pow((Acc_rawZ/16384.0),2)))*rad_to_deg;

```

```

/*Now we read the Gyro data in the same way as the Acc data. The adress for the
* gyro data starts at 0x43. We can see this addresses if we look at the register map
* of the MPU6050. In this case we request just 4 values. W don;t want the gyro for
* the Z axis (YAW).*/

```

```

Wire.beginTransmission(0x68);
Wire.write(0x43); //Gyro data first adress
Wire.endTransmission(false);
Wire.requestFrom(0x68,4,true); //Just 4 registers

Gyr_rawX=Wire.read()<<8|Wire.read(); //Once again we shif and sum
Gyr_rawY=Wire.read()<<8|Wire.read();

/*Now in order to obtain the gyro data in degrees/seconda we have to divide first
the raw value by 131 because that's the value that the datasheet gives us*/

/*---X---*/
Gyro_angle[0] = Gyr_rawX/131.0;
/*---Y---*/
Gyro_angle[1] = Gyr_rawY/131.0;

/*Now in order to obtain degrees we have to multiply the degree/seconds
*value by the elapsedTime.*
/*Finnaly we can apply the final filter where we add the acceleration
*part that affects the angles and ofcourse multiply by 0.98 */

/*---X axis angle---*/
Total_angle[0] = 0.98 *(Total_angle[0] + Gyro_angle[0]*elapsedTime) +
0.02*Acceleration_angle[0];
/*---Y axis angle---*/
Total_angle[1] = 0.98 *(Total_angle[1] + Gyro_angle[1]*elapsedTime) +
0.02*Acceleration_angle[1];

/*Now we have our angles in degree and values from -10°0 to 100° aprox*/
//Serial.println(Total_angle[1]);

```

```

/*////////////////////////////////P I D////////////////////////////////*/
/*Remember that for the balance we will use just one axis. I've choose the x angle
to implement the PID with. That means that the x axis of the IMU has to be paralel to
the balance*/

/*First calculate the error between the desired angle and
*the real measured angle*/
error = Total_angle[1] - desired_angle;

/*Next the proportional value of the PID is just a proportional constant
*multiplied by the error*/

pid_p = kp*error;

/*The integral part should only act if we are close to the
desired position but we want to fine tune the error. That's
why I've made a if operation for an error between -2 and 2 degree.
To integrate we just sum the previous integral value with the
error multiplied by the integral constant. This will integrate (increase)
the value each loop till we reach the 0 point*/
if(-3 <error <3)
{
  pid_i = pid_i+(ki*error);
}

/*The last part is the derivate. The derivate acts upon the speed of the error.
As we know the speed is the amount of error that produced in a certain amount of
time divided by that time. For taht we will use a variable called previous_error.
We substract that value from the actual error and divide all by the elapsed time.
Finnaly we multiply the result by the derivate constant*/

pid_d = kd*((error - previous_error)/elapsedTime);

/*The final PID values is the sum of each of this 3 parts*/
PID = pid_p + pid_i + pid_d;

/*We know taht the min value of PWM signal is 1000us and the max is 2000. So that
tells us that the PID value can/s oscilate more than -1000 and 1000 because when we
have a value of 2000us the maximum value taht we could sybstract is 1000 and when
we have a value of 1000us for the PWM sihnal, the maximum value that we could add
is 1000
to reach the maximum 2000us*/

```

```

if(PID < -1000)
{
  PID=-1000;
}
if(PID > 1000)
{
  PID=1000;
}

```

*/\*Finally we calculate the PWM width. We sum the desired throttle and the PID value\*/*

```

pwmLeft = throttle + PID;
pwmRight = throttle - PID;

```

*/\*Once again we map the PWM values to be sure that we won't pass the min and max values. Yes, we've already mapped the PID values. But for example, for throttle value of 1300, if we sum the max PID value we would have 2300us and that will mess up the ESC.\*/*

```

//Right
if(pwmRight < 1000)
{
  pwmRight= 1000;
}
if(pwmRight > 2000)
{
  pwmRight=2000;
}
//Left
if(pwmLeft < 1000)
{
  pwmLeft= 1000;
}
if(pwmLeft > 2000)
{
  pwmLeft=2000;
}

```

*/\*Finally using the servo function we create the PWM pulses with the calculated width for each pulse\*/*

```

left_prop.writeMicroseconds(pwmLeft);
right_prop.writeMicroseconds(pwmRight);
previous_error = error; //Remember to store the previous error.

```

```

} //end of loop void

```



## PROFIL PENULIS



Alfius Christian Katilik, lahir di Bandar Lampung, Lampung pada tanggal 9 April 1998 merupakan anak pertama dari 3 bersaudara. Penulis menempuh pendidikan tingkat kanak kanak di TK Xaverius Panjang, Lampung, melanjutkan pendidikan tingkat sekolah dasar di SD Xaverius Panjang kemudian SD Seruni Putih Pamulang. Dilanjutkan di SMPN 17 Tangerang Selatan, kemudian SMAN 3 Tangerang Selatan pada jalur akselerasi. Setelah lulus dari bangku SMA, penulis melanjutkan pendidikan perguruan tinggi di Departemen Teknik Sistem Perkapalan, Fakultas Teknologi Kelautan, Institut Teknologi Sepuluh Nopember Surabaya.

Selama Menempuh Pendidikan di Departemen Teknik Sitem Perkapalan ITS, penulis aktif dalam berbagai bidang kegiatan dan organisasi kampus. Pada Tahun 2015/2016 penulis aktif dalam kegiatan kepanitian diantaranya kegiatan panitia Petroleum Integrated Days SPE ITS dan Seminar Nasional POSEIDON BEM ITS. Di periode tahun 2016/2017 sampai tahun 2018/2019, penulis aktif dalam kegiatan kerohanian di Persekutuan Mahasiswa Kristen ITS baik sebagai staf, panitia acara, staff ahli, mentor maupun pemusik.

Pada periode tahun yang sama penulis juga mejadi bagian dari tim riset robot kapal tanpa awak yaitu Tim Barunastra ITS sebagai anggota divisi mekanik. Bersama Barunstra, penulis berhasil mendapatkan beberapa gelar penghargaan baik tingkat nasional maupun internasional diantaranya Juara 1 Nasional KKCTBN 2016, Juara 1,2 dan Best Design Deconbution UNDIP 2017, Juara 4 International Roboboat Competition 2017 di Florida, USA serta juara 1 International Roboboat Competition 2018 yang diadakan oleh Robonation

Email : [alfiuschk@gmail.com](mailto:alfiuschk@gmail.com)