



**RANCANG BANGUN SISTEM WATER BALLAST
OTOMATIS BERBASIS ARDUINO R3**

SKRIPSI

**Untuk memperoleh gelar Sarjana Terapan Pelayaran pada
Politeknik Ilmu Pelayaran Semarang**

Oleh

FARIZ FAUZIAN

551811236931 T

PROGRAM STUDI TEKNIKA DIPLOMA IV

POLITEKNIK ILMU PELAYARAN

SEMARANG

TAHUN 2023

HALAMAN PERSETUJUAN

**RANCANG BANGUN SISTEM WATER BALLAST
OTOMATIS BERBASIS ARDUINO R3**

Disusun Oleh :

FARIZ FAUZIAN

551811236931 T

Telah disetujui dan diterima, selanjutnya dapat diujikan didepan Dewan Penguji
Politeknik Ilmu Pelayaran Semarang . . .

Dosen Pembimbing I
Materi

Dosen Pembimbing II
Metodelogi dan Penulisan

AMAD NARTO, M.Mar.E., M.Pd.

Pembina (IV/a)

NIP. 19641212 199808 1 001

ELY SULISTIOWATI, S.ST., M.M

Penata Tk. I (III/d)

NIP. 197870801200612 2 001

Mengetahui
Ketua Program Studi Teknika

AMAD NARTO, M.Mar.E., M.Pd.

Pembina (IV/a)

NIP. 19641212 199808 1 001

PENGESAHAN UJIAN SKRIPSI

Skripsi dengan judul “Rancang Bangun sistem *water ballast otomatis*” karya,

Nama : Fariz Fauzian

NIT : 551811236931 T

Program Studi : Teknika

Telah dipertahankan di hadapan Panitia Penguji Skripsi Prodi Teknika, Politeknik Ilmu Pelayaran Semarang pada hari, tanggal

Semarang, Juli 2023

Panitia Ujian

Dr. ANDY WAHYU HERMATO, M.T.

Peneta Tk. I (III/d)

NIP. 19791212 200012 1 001

H. AMAD NARTO. M.Mar.E., M.Pd

Pembina (IV/a)

NIP. 19641212 199808 1 001

PURWANTONO, S.Psi., M.Pd

Penata Tk. I (III/d)

NIP. 196610151997703 1 002

Mengetahui
Direktur Politeknik Ilmu Pelayaran Semarang

Dr. Capt. TRI CAHYADI, M.H., M.Mar.

Pembina Tk. I (IV/b)

NIP. 19730704 199803 1 001

PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini :

Nama : Fariz Fauzian

NIT : 551811236931 T

Program Studi : Teknika

Skripsi dengan judul “Rancang Bangun sistem *water ballast otomatis* berbasis Arduino R3”

Dengan ini saya menyatakan bahwa yang tertulis dalam skripsi ini benar-benar hasil karya (penelitian dan tulisan) sendiri, bukan jiplakan dari karya tulis orang lain atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku, baik Sebagian atau seluruhnya. Pendapat atau temuan orang lain yang terdapat dalam skripsi ini dikutip atau dirujuk berdasarkan kode etik ilmiah. Atas pernyataan ini saya siap menanggung resiko/sanksi yang dijatuhkan apabila ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya ini.

Semarang, 2023
Yang menyatakan

FARIZ FAUZIAN
NIT. 551811236931T

Moto dan Persembahan

Moto

1. “Dan bersabarlah kamu, sesungguhnya janji Allah adalah benar.” (Q.S Arum:60).
2. “Kemudian apabila kamu telah membulatkan tekad, maka bertakwalah kepada Allah. Sesungguhnya Allah menyukai orang-orang yang bertawakal kepada nya” (Q.S Alimran:159).
3. “Barang siapa yang mengerjakan kebaikan sekecil apapun, niscaya dia akan melihat (balasan)nya” (Q.S Al-Zazalah:7).



Persembahan:

1. Orang tua peneliti, Bapak Suparno.SE dan Ibu Iis Susilawati.
2. Geia aisyah kusumawati yang selalu memberikan support dan semangat.
3. Dosen Pembimbing I Bapak H. Amad Narto, M.Mar.,M.pd. dan Dosen Pembimbing II Ibu Ely Sulistiowati, S,ST.,M.M.

PRAKATA

Alhamdulillah, segala puji syukur saya panjatkan kehadirat Allah SWT, yang Maha Pengasih lagi Maha Penyayang atas segala rahmat dan hidayah-Nya yang telah dilimpahkan kepada hamba-Nya sehingga Skripsi ini dapat terselesaikan dengan baik. Sholawat serta salam senantiasa tercurahkan kepada Nabi Muhammad SAW yang telah mengantarkan kita menuju jalan yang benar.

Skripsi ini mengambil judul “Rancang Bangun sistem *water ballast otomatis*” yang telah terselesaikan berdasarkan data-data yang diperoleh dari hasil penelitian selama satu tahun lima belas hari praktek laut di perusahaan PT. Salam Pacifik Indonesia Line.

Dalam usaha menyelesaikan Penulisan Skripsi ini, dengan penuh rasa hormat Penulis menyampaikan ucapan terimakasih kepada pihak-pihak yang telah memberikan bimbingan, dorongan, bantuan serta arahan yang berarti. Untuk itu pada kesempatan ini Penulis menyampaikan ucapan terimakasih kepada yang terhormat :

1. Bapak Dr. Capt. Tri Cahyadi, M.H., M.Mar., selaku Direktur Politeknik Ilmu Pelayaran Semarang yang telah memberikan kemudahan dalam menuntut ilmu di Politeknik Ilmu Pelayaran Semarang.
2. Bapak H. Amad Narto, M.Mar.,M.Pd. selaku Ketua Program Studi Teknika Politeknik Ilmu Pelayaran Semarang yang telah memberikan kemudahan dalam menuntut ilmu di Politeknik Ilmu Pelayaran Semarang.

3. Bapak H. Amad Narto, M.Mar.E.,M.Pd. selaku Dosen Pembimbing Materi Penulisan Skripsi yang dengan sabar dan tanggung jawab telah memberikan dukungan, bimbingan, dan pengarahan dalam penyusunan Skripsi ini.
4. Ibu Ely Sulistiowati, S, ST.,M.M selaku Dosen Pembimbing Metode Penulisan Skripsi yang telah memberikan dukungan, bimbingan, dan pengarah dalam penyusunan Skripsi ini. Pimpinan beserta Karyawan Perusahaan PT. Salam Pacifik Indonesia Line. yang telah memberikan kesempatan pada Penulis untuk penelitian dan praktek di atas kapal.
5. Nahkoda, KKM beserta seluruh awak MV. SPIL RUMI yang telah membantu Penulis dalam melaksanakan penelitian dan praktek.
6. Bapak dan Ibu tercinta, serta Kakak dan adik yang telah memberikan dukungan moral dan spiritual kepada Penulis selama Penulisan Skripsi ini.
7. Semua pihak dan rekan-rekan yang telah memberikan motivasi serta membantu Penulis dalam penyusunan Skripsi ini.

Akhirnya, dengan segala kerendahan hati Penulis menyadari masih yang banyak terdapat kekurangan-kekurangan, sehingga Penulis mengharapkan adanya saran dan kritik yang bersifat membangun demi kesempurnaan Skripsi ini bermanfaat bagi seluruh pembaca

Semarang,
Penulis

FARIZ FAUZIAN
551811236931 T

ABSTRAKSI

Fauzian, Fariz. 2023. “*rancang bangun sistem water ballast otomatis untuk peningkatan teknologi dibidang pelayaran dan pembelajaran bagi taruna*”. skripsi. Program Diploma IV, Program studi Teknika, Politeknik Ilmu Pelayaran Semarang, Pembimbing I: H. Amad Narto, M.Mar.E.,M.Pd. pembimbing II: Ely Sulistiowati, S, ST.,M.M

Di era modern ini perkembangan teknologi sangatlah pesat dan banyak perubahan yang terjadi dibidang teknologi khususnya dikehidupan manusia, yang tentunya hal ini juga berlaku dibidang maritim. Teknologi merupakan sesuatu yang sangat penting untuk memudahkan urusan pada aktifitas dalam kehidupan manusia sehingga segala sesuatu yang ada mampu diselesaikan dengan cepat, mudah dan efektif. Tentunya perkembangan teknologi ini sangatlah banyak kemampuan dalam mejalani prasarana penting untuk meningkatkan kualitas kerja dan efisiensi waktu. Teknologi juga sangat bermanfaat pada sector bisnis, karena dengan ada nya perkembangannya teknologi orang-orang akan semakin mudah mengenal satu dengan lainnya.

Penelitian ini bertujuan untuk mengetahui bagaimana cara kerja *sistem pengisian ballast* serta mengetahui komponen-komponen yang berada pada alat tersebut. Motode yang digunakan adalah metode *Research & Development (R&D)*.

Berdasarkan temuan dan hasil penelitian tersebut, didapatkan kesimpulan bahwa *sistem ballast* adalah alat yang digunakan untuk menambah kestabilan pada kapal dan menambah gaya G untuk mudah dikenadalikan pada titik keseimbangan structure kapal yang bermuatan maupun tidak bermuatan. Perancangan alat peraga pengoprasian *sistem water ballast* dilakukan secara urut pada setiap proses pembuatannya dan sudah disesuaikan dengan rumusan masalah. Alat peraga ini sangat bermanfaat sebagai media pembelajaran sehingga materi lebih mudah dipahami karena diperagakan dengan menggunakan sebuah alat peraga yang nyata.

Kata kunci: *water ballasts auto, prototype, media pembelajaran*

ABSTRACT

Fauzian, Fariz. 2023. "Design an Automatic Water Ballast System for Technology Improvement in the Field of Sailing and Learning for Cadets". Thesis. Diploma IV Program, Engineering Study Program, Semarang Shipping Science Polytechnic, Supervisor I: H. Amad Narto, M.Mar.E.,M.Pd. Supervisor II: Ely Sulistiowati, S, ST., M.M

In this modern era, technological development is very rapid and many changes occur in the field of technology, especially in human life, which of course also applies in the maritime field. Technology is something that is very important to facilitate affairs in activities in human life so that everything that exists can be completed quickly, easily and effectively. Of course, the development of this technology is very much the ability to carry out important infrastructure to improve work quality and time efficiency. Technology is also very useful in the business sector, because with the development of technology people will more easily get to know each other.

This study aims to find out how the ballast filling system works and find out the components that are in the tool. The mode used is the Research & Development (R&D) method.

Based on the findings and results of the study, it was concluded that the ballast system is a tool used to add stability to the ship and increase the G force to be easily controlled at the balance point of the structure of the ship that is fast or unloaded. The design of water ballast system operating props is carried out sequentially in each manufacturing process and has been adjusted to the problem formulation. This teaching aid is very useful as a learning medium so that the material is easier to understand because it is demonstrated using a real teaching aid.

Keywords: water ballats auto, prototype, learning media

DAFTAR ISI

Halaman Judul	I
Halaman Persetujuan	II
Halaman Pengesahan	III
Halaman Pernyataan Keaslian	IV
Halaman Moto dan Persembahan	V
Prakata	VI
Abstraksi	VIII
Abstract	IX
Daftar Isi	X
Daftar Tabel	XIII
Daftar Gambar	XIV
Daftar Lampiran	XVI
Bab I Pendahuluan	1
Latar Belakang	1
Rumusan Masalah	4
Tujuan Penelitian	4
Manfaat Hasil Penelitian	6
Bab II Landasan Teori dan Pengajuan Hipotesis	9
Dekripsi Teori	9
Kerangka Berfikir	27
Hipotesis	28

Bab III Metode Penelitian	30
Langkah-langkah Penelitian	30
Metode Penelitian Tahap I (<i>Research</i>)	30
Metode Kualitatif	31
Tempat Penelitian	31
Sumber Data Penelitian	32
Teknik Pengumpulan Data	32
Analisis Data	32
Perencanaan Desain Produk	33
Validasi Desain	34
Metode Penelitian II (<i>Development</i>)	34
Desain Uji Produk	35
Subjek Penelitian	38
Teknik Pengumpulan Data	38
Observasi	39
Instrumen Penelitian	39
Uji keabsahan data	40
Lembar angket respon taruna	40
Teknik Analisis Data	41
Reduksi data	41
Penyajian data	42
Kesimpulan atau verifikasi	45

Bab IV Hasil Penelitian	46
Desain Awal Produk	46
Hasil Pengujian Pertama	50
Revisi Produk	53
Hasil Pengujian Tahap ke II	55
Revisi Produk	61
Pengujian Tahap Ke III.....	62
Penyempurnaan Produk	70
Pembahasan Produk	71
Bab V Simpulan dan Saran	80
Simpulan	80
Saran	81
Daftar Pustaka	
Lampiran	
Daftar Riwayat Hidup	

DAFTAR TABEL

Tabel 3.1 Respon	43
Table 3.2 Nilai Respon	43
Table 3.3 Darftar Pertanyaan	43
Table 3.4 Skala dan Kategori Kelayakan	45



DAFTAR GAMBAR

Gambar 2.1 adaptor 6v	12
Gambar 2.2 Arduino R3.....	13
Gambar 2.3 LCD Ic2.....	15
Gambar 2.4 kabel jumper	16
Gambar 2.5 MPU 6050.....	17
Gambar 2.6 sensor <i>water level</i>	18
Gambar 2.7 pompa air 6-12v	19
Gambar 2.8 lampu LED 1,5v.....	20
Gambar 2.9 saklar	22
Gambar 2.10 Buzzer	24
Gambar 2.11 <i>Step Down 12V</i>	25
Gambar 2.12 <i>Relay</i>	26
Gambar 4.1 desain <i>electronic level hookup</i>	48
Gambar 4.2 strukture <i>ballast tank</i>	49
Gambar 4.3 strukture kapal	50
Gambar 4.4 proses pemotongan dan perbaikan buritan.....	51
Gambar 4.5 proses pendempulan buritan	51
Gambar 4.6 proses pelapisan cat	52
Gambar 4.7 produk cat saporo	53
Gambar 4.8 peletakan posisi pipa <i>ballast</i>	56
Gambar 4.9 <i>plane in and out ballast pipe line</i>	57
Gambar 4.10 <i>plane in and out ballast pipe line</i>	58

Gambar 4.11 <i>plane in and out ballast pipe line</i>	58
Gambar 4.12 uji <i>ballast tank</i>	59
Gambar 4.13 <i>pump room</i>	60
Gambar 4.14 hasil rancang bangun	70
Gambar 4.15 hasil Bersama validasi rancang bangun	79



DAFTAR LAMPIRAN

Lampiran 1 Kuesioner	84
Lampiran 2 Bukti Data	86
Lampiran 3 Lembar Validasi Ahli	89
Lampiran 4 Program Coding Sensor	91
Lampiran 5 <i>Manual Book</i>	110



BAB I

PENDAHULUAN

A. Latar Belakang

Perkembangan teknologi sangatlah pesat dalam kehidupan manusia hal ini juga berlaku dalam bidang maritim, teknologi merupakan suatu hal yang sangat penting bagi kehidupan manusia untuk mempermudah suatu pekerjaan berat maupun ringan sehingga segala sesuatu dapat diselesaikan dengan cepat, efektif, dan efisien. Tentunya perkembangan teknologi ini dapat membuat perubahan besar Bagi umat manusia dan mempunyai peranan penting untuk meningkatkan kualitas kerja dan efektivitas waktu yang digunakan. Teknologi juga Sangat bermanfaat pada bidang bisnis, karena perkembangan teknologi Orang-orang akan mengenal dengan mudah satu dengan lainnya. Kita juga bisa memilih mana yang berdampak positif dan negatif, karena setiap Manusia bebas memilih haknya masing-masing untuk memilih. Manusia yang cerdas pasti akan bisa memanfaatkan adanya perkembangan Teknologi bahkan bisa menjadikan suatu kemajuan teknologi lainnya dan Ini menjadi ladang pencarian dan penghasilan. Perkembangan tersebut Sangat berdampak positif dalam kemajuan perindustrian sehingga lebih Efisien dan praktis. Maka dari itu permesinan dibidang maritimpun Sangat berkembang pesat yang berguna untuk mempermudah Pengoperasian dikapal dan efektivitas dalam bekerja.

Berkembangnya ilmu dan pengetahuan didunia maritim akan semakin diterapkan pada dunia kerja. Cepat atau lambat berbagai macam ilmu dan pengetahuan tentang teknologi semakin digali supaya terciptanya lingkungan

kerja yang lebih memudahkan dalam menggunakannya. Awal mula teknologi tercipta itu bersal dari pemikiran dan perkembangan umat manusia yang berevolusi, akan tetapi hal itu justru akan memicu semua orang untuk selalu meningkatkan keahlian dan potensi yang ada pada diri supaya tidak tertinggal zaman pada perkembangan teknologi. Contoh kecilnya adalah sistem kontrol otomatis Bergantinya zaman, sistem kontrol otomatis sudah mulai Berkembang dikehidupan manusia, dimana sistem kontrol adalah proses pengaturan atau pengendalian terhadap satu atau beberapa besaran (variabel, parameter) sehingga berada pada suatu harga atau range tertentu. Sistem kontrol atau teknik kendali adalah Teknik pengaturan, sistem pengendali, atau sistem pengontrolan Thomas,(1987). Contoh sistem kontrol di atas kapal adalah penggunaan *system water ballast*. Menurut Jhson, F.(2002) sistem *ballast* adalah salah satu sistem penyeimbang muatan pelayaran di kapal yang dimana *ballast* akan menambah dan membuang volume air yang berada ditanki *ballast*.

Sistem *ballast* berfungsi untuk memposisikan kapal menjadi stabil dalam keadaan kosong dan bermuatan, perlu untuk mengembangkan sebuah rancang bangun yang berguna sebagai media pembelajaran bagi taruna/mahasiswa teknik terutama dibidang maritim untuk mempelajari tentang pengoprasian *system water ballast otomatis*. Agar lebih mudah untuk dimengerti pengoperasiannya diatas kapal. Sistem ini digunakan untuk menyeimbangkan kapal, air akan mengisi disetiap tangki- tangka *ballast* yang diorder oleh modul yang membaca pergerakan atau kemiringan kapal. Berdasarkan pengetahuan yang telah diterapkan, sistem ini sangat diperlukan

untuk menunjang pengoperasian *system water ballast otomatis*. Pada sistem perancangan adalah menggunakan alat sederhana yang mudah diaplikasikan dalam kehidupan sehari-hari.

Sesuai yang diterapkan dalam *ballast water management* bahwa penggunaan *ballast* untuk mengontrol stabilitas kapal. Dalam arti lain *ballast water* adalah air yang digunakan untuk mengontrol stabilitas kapal yang tak bermuatan. Aturan yang meliputi lokasi pergantian *ballast water* seperti jarak yang ditentukan dari daratan ke lautan harus melebihi *12 nautical mile* dan kedalaman laut melebihi 200 meter serta densitas air lautnya. Aturan tersebut diatur dalam *ballast water management convention and BWMS code*.

Konvensi IMO, (*internasional maritime organization*) sebutan resminya adalah "*The International Convention for the Control and Management of Ships' Ballast Water and Sediments*" mulai diadopsi IMO pada tahun 2004 dan mulai diberlakukan (*Entry into force*) pada 8 September 2017. Perlu kita pahami disetiap negara memiliki aturan dan peraturan lautnya masing-masing untuk menjaga kelestarian ekosistem dan habitat ataupun kekayaan laut lainnya yang ada disetiap negara mereka. Regulasi ini sangat berdampak positif bagi semua dan ekosistem, karena setiap manusia pasti ingin terciptanya kondisi lingkungan yang bersih dan tidak mencemari lingkungan kehidupan di muka bumi ini. Lingkungan laut merupakan lingkungan perairan yang harus kita jaga kelestariannya agar dimasa depan nanti anak dan cucu kita. Penelitian ini diharapkan mampu memberikan dampak yang sangat positif pengetahuan dan pembelajaran untuk mempermudah memahami prinsip kerja *system water ballast otomatis*. Selain itu penelitian ini diharapkan bisa berkembang lagi

untuk menunjang teknologi ini supaya lebih canggih, moderen dan bermanfaat bagi pembelajaran. Penelitian ini menggunakan ide, wawasan dan gagasan melalui pengembangan model dengan judul “*sistem pengisian water ballast otomatis*” untuk meningkatkan teknologi dibidang maritim dan pembelajaran bagi taruna/mahasiswa.

B. Fokus penelitian

Tahap ini digunakan oleh peneliti untuk menganalisis kinerja dari sistem *water ballast* dan di terapkan dengan ada nya ide dari sebuah pemikiran peneliti yang akan membuat operasiona kapal menjadi efisien. Penelitian ini akan berfokus pada pengembangan dan peningkatan teknologi *water ballast* otomatis. Peneliti dapat mencari cara untuk meningkatkan sistem kontrol otomatis, serta menganalisis bagaimana teknologi ini berkontribusi pada pengurangan dampak negatif terhadap sistem atau kinerja yang akan diterapkan

C. Rumusan masalah

Perumusan masalah dalam penelitian ini adalah:

1. Apa saja proses dalam pembuatan *water ballast* otomatis?
2. Komponen apa saja yang digunakan untuk pembuatan *water ballast* otomatis?
3. Bagaimana cara proses *water ballast otomatis* bekerja?

D. Tujuan penelitian

Berdasarkan rumusan masalah yang telah dipaparkan, peneliti mengambil beberapa tujuan penelitian. Tujuan penelitan tersebut antara lain:

1. Proses pembuatan *water ballast* otomatis ini melibatkan beberapa tahapan seperti :

- a. Perancangan sketsa rancang bangun *water ballast* otomatis.
 - b. Memilih material yang mudah dicari.
 - c. Proses pemolesan dan pewarnaan pada bagian kapal.
 - d. Pemilihan sensor elektronik yang akan digunakan.
 - e. Pemilihan posisi peletakan sensor yang akan diaplikasikan ke rancang bangun *water ballast* otomatis agar struktur kapal seimbang.
 - f. Pemotongan dan pengukuran bahan yang akan digunakan
 - g. Mengimput data yang akan digunakan pada sensor Arduino R3 dan sensor lainnya.
2. Peneliti menggunakan beberapa komponen elektrikal dan sensor pada rancang bangun *water ballast* otomatis tahap pemilihan komponen ini bertujuan untuk meminimalisir kesalahan yang terjadi akibat adanya komponen error atau tidak sesuai dengan spesifikasi yang telah ditentukan oleh peneliti. Komponen kelistirikan yang tidak dapat di sinkronkan dengan komponen lainya dapat membuat program tidak berjalan dengan semestinya.
3. Proses *water ballast* otomatis akan berkerja dengan cara mengisi dan memnbuang air yang berada di dalam tangki *ballast* secara otomatis, ketika masa kapal diberikan beban dibagian kanan maupun disebelah kiri dan sensor akan membaca pergerakan pada sudut kemiringan (*Roll*) yang terjadi pada kapal dan sensor akan bekerja untuk mengembalikan possisi gyro 0 derajat agar kondisi kapal tetap seimbang dan stabil.

D. Manfaat hasil penelitian

Perkembangan sistem *water ballast* diteknologi perkapalan memiliki dampak positif untuk sebuah keuntungan dari berbagai aspek seperti: memudahkan pengoperasian *water ballast*, efisiensi waktu dan bagi *crew* kapal memiliki tenaga dan dapat menghasilkan tingkat keselamatan yang tinggi serta mampu mempermudah pekerjaan. Pada riset ini peneliti mengembangkan pengoperasian *water ballast*, menggunakan sistem elektronika sebagai teknologi pendamping di kapal untuk mempermudah perkembangan sistem dan pengoperasian tidak membutuhkan tenaga terlalu banyak untuk pengoperasian alat.

1. Manfaat teoritis

a. Bagi peneliti

Penelitian ini bisa menjadi peluang untuk memperbesar pengetahuan dan berinovasi untuk mengembangkan sistem control di atas kapal agar lebih mempermudah penggunaan pada sistem *control water ballast*.

b. Bagi Lembaga Pendidikan

Hasil dari pengembangan sistem control memiliki manfaat yaitu menambah media pembelajaran yang ada di kampus Politeknik Ilmu Pelayaran Semarang dan menjadi motivasi untuk Taruna dan Taruni dalam mengembangkan kreativitasnya dalam bidang teknologi dan dapat menambah pengetahuan mahasiswa maupun mahasiswa di dunia teknologi. Hasil karya ini bermanfaat juga meningkatkan akreditasi kampus dan mempertahankan kualitas pendidikan.

c. Bagi pembaca

Wawasan dan pengetahuan baru tentang pengoperasian *water ballast* dapat menjadi acuan untuk peneliti maupun pengembang serta dapat menjadi pembelajaran untuk diterapkan pada dunia kerja. Penelitian ini memiliki manfaat mampu meningkatkan wawasan dan pengetahuan pada bidang pelayaran supaya mahasiswa lebih mudah memahami bagai mana perkembangan teknologi yang sudah ada dan mampu diterapkan. Perkembangan teknologi mengakibatkan manusia harus berfikir cerdas untuk menghadapi teknologi dikarenakan masalah tuntutan pekerjaan.

2. Manfaat Praktis

a. Dalam industri maritim

Teknologi ini sangat penting bagi keamanan kapal dengan adanya sistem *water ballast* otomatis, pengoperasian sistem *ballast* menjadi lebih mudah dan efisien.

Karena pada dasarnya sistem ini bekerja dengan cara otomatis untuk menjaga stabilitas dan keamanan kapal agar tetap stabil disilain kegunaan alat ini juga sangata penting untuk menghemat biaya dan waktu karena sistem kontrol *water ballast* otomatis ini dapat mengatur ballast air dengan lebih cepat dan efisien.





BAB II

LANDASAN TEORI

A. Deskripsi Teori

Pada bab ini landasan teori dipergunakan untuk sebuah teori yang menjadikan sumber dasar sebuah penelitian. Sumber tersebut memperoleh dasar atau kerangka untuk lebih memahami latar belakang pengembangan ini.

1. Desain alat peraga

Pembuatan rancang bangun *sistem water ballast otomatis*, membutuhkan proses untuk memberikan informasi tahap perancangan yang dapat memberikan gambaran dan bertujuan untuk memberikan kemudahan pengguna untuk membuat model atau bentuk dari sebuah sistem elektronik dan *sistem water ballast otomatis* yang akan dibuat serta menggabungkan beberapa elemen yang terpisah sehingga bisa menjadi satu dan memberikan fungsi yang sebanding dengan apa yang dikehendaki. Pada tahap ini peneliti mempersiapkan desain alat peraga pengoprasian *sistem water ballast otomatis* agar tercipta rancangan yang tepat dan harus dipertimbangkan ukuran dan bahan agar alat peraga dapat berfungsi dengan baik dan memiliki bentuk yang ideal sehingga tercipta alat peraga yang ekonomis.

2. Sistem kendali manual

Teknologi kendali manual merupakan teknologi yang berhubungan dengan model pengoprasian sehingga campur tangan manusia ada dalam pengoprasian, sistem kendali ini tidak mempergunakan sensor yang memerintahkan alat untuk bekerja. Dalam sistem pengendali manual, sistem ini tetap menggunakan komponen elektronika untuk menjalankan

perintah yang peneliti inginkan. Pengendali lokal adalah Sebagian pengendali operator, yaitu bagian dimana pengontrol diberikan perintah akses pada pengendalinya, sedangkan bagian pada sisi otomatis yaitu bagian yang berhubungan langsung menggunakan alat yang dapat dikendalikan.

3. Maket

Menurut penuturan Khairi (2012: 13), maket diartikan sebagai bentuk tiga dimensi yang bisa dipakai dalam dunia arsitektur, menurut penuturan maket diartikanya sebagai benda tiga dimensional berbentuk tiruan benda atau suatu objek seperti gedung, pesawat, kapal, dan lainnya dibuat dengan skala yang lebih kecil dan biasa dibuat menggunakan kayu tanah dan bahan lainnya. Jadi maket yaitu sebagai bentuk tiruan seperti gedung, pesawat terbang, kapal, dan sebagainya dalam tiga dimensi dan berskala kecil. Dalam pembuatan alat peraga ini peneliti membuat miniatur alat peraga dalam bentuk sistem ballast dengan menggunakan bahan serat viber, resin sebagai penegas dan akrilik.

Dalam teori teori tersebut maket dapat diartikan sebagai dasain tiruan yang terbuat dalam skala kecil dan berbentuk tiga dimensi. Media pada maket juga merupakan media bagi pembelajaran dengan metode media visual tiga dimensi yang dapat berguna untuk menampilkan gambaran visual mengenai bangunan atau area yang dimaksud dengan skala yang tepat, tanpa perlu merubah objek aslinya, dan dalam hal ini peneliti merangkai *miniature sistem water ballast otomatis*.

4. Perencanaan dan pemrograman pengoprasian *sistem water ballast otomatis*

Dalam perancangan hampir semua alat pengoprasian dikapal

menggunakan sistem otomatis, sebuah alat yang bergerak dikontrol oleh sebuah sistem sensor. Sensor adalah sebuah elemen yang berfungsi sebagai media pembaca sebuah sinyal arus rendah yang akan meneruskan perintah ke *point control* atau juga sebagai saklar otomatis ketika terdeteksi sebuah benda, cairan, ataupun sinyal yang memicu reaksi sensor. Pada sebuah alat sistem *water ballast* digunakan sistem kendali secara manual karena untuk pengoperasian ballast membutuhkan pengoperasian dari *crew* yang bekerja ataupun yang sedang menjalankan dinas jaga sehingga mencapai tujuan yang diharapkan.

Untuk menopang rancangan pada alat peraga ini tentunya peneliti membutuhkan *part-part* yang sesuai dengan elektronik yang dibutuhkan.

Komponen elektronik yang harus disiapkan antara lain:

a. Adaptor

Untuk mempermudah dalam pengoperasian rancang bangun sistem *water ballast* otomatis maka peneliti memperbanyak perangkat elektronik. Contohnya adaptor. Adaptor berfungsi untuk mengubah tipe arus listrik bolak-balik dengan nilai yang tinggi menjadi tegangan listrik tipe arus rendah dengan nilai yang rendah. Seperti kita ketahuhi bahwa halnya arus listrik yang kita gunakan dirumah, kantor, sekolah, tempat swalayan dll adalah arus listrik dari PLN yang di distribusikan dalam bentuk arus bolak-balik yang akan digunakan. Elektronik yang kita gunakan Sebagian besar membutuhkan arus searah DC dengan arus yang rendah untuk sistem pengoperasiannya. Diperlukan sebuah modul atau alat dengan rangkaian yang berfungsi untuk merubah arus bolak-

balik menjadi searah serta diberikan tegangan sesuai dengan kebutuhannya. Rangkaian yang berfungsi merubah arus AC menjadi DC ini disebut dengan kata lain *power supply* atau di sebut juga dengan adaptor. Rangkaian adaptor ini ada yang terpisah dan juga ada terpasang langsung dengan komponen elektroniknya. Adaptor yang terpisah dari komponen modul yaitu adaptor *universal* yang mempunyai tegangan *Volt* yang sesuai kebutuhan yaitu 3V, 4,5V, 6V, 9V, 12V dan beraneka ragam. Ada juga adaptor yang menyediakan besar tegangan tertentu dan disesuaikan dengan kapasitas elektroniknya contohnya yaitu adaptor handphone dan adaptor laptop.



Gambar 2.1 adaptor 6V
(sumber: dokumen pribadi)

b. Arduino R3

Arduino adalah perangkat purwarupa (*prototyping*) yang bersifat *open source*. Merupakan sebuah perangkat elektronik dan sering digunakan untuk merancang dan membuat perangkat elektronik serta *software* yang mudah untuk digunakan. Arduino dirancang sedemikian

rupa untuk mempermudah pengguna perangkat elektronik diberbagai bidang. Arduino ini memiliki beberapa komponen penting didalamnya seperti pin, mikrokontroler, dan konektor yang nanti akan digunakan. Arduino juga sudah menggunakan bahasa pemrograman arduino *language* yang sedikit mirip dengan bahasa pemrograman C++.

Biasanya Arduino digunakan untuk mengembangkan beberapa sistem seperti pengatur suhu, sensor untuk dibidang *agriculture*, pengendali peralatan pintar dan masih banyak lagi.



Gambar 2.2 Arduino R3
(sumber: dokumen pribadi)

c. LCD ic2

LCD (*liquid crystal display*) IC2 module adalah modul yang dikendalikan secara serial sinkron dengan protocol I2C/IIC (*inter integrated circuit*) atau TWI (*two wire interface*). Normal nya modul

LCD dikendalikan secara *parallel* baik untuk jalur data maupun control nya. Pada modul LCD ic2 memiliki beberapa kegunaannya di antaranya:

1. Mengontrol tampilan: IC ini berfungsi mengontrol tampilan pada modul LCD, baik itu menampilkan teks, angka, maupun gambar.
2. Komunikasi dengan komponen lainnya: IC ini merupakan modul LCD untuk terhubung dengan sistem komponen lainnya dalam sebuah sistem melalui protocol I2C. sebagai contoh, IC2 pada modul LCD dapat digunakan untuk mengirim dan menerima data dari mikrokontroler atau prosesor lain dalam sebuah sistem.
3. Meningkatkan kemampuan modul LCD: modul LCD dengan IC2 memiliki kemampuan tambahan seperti kemampuan *touchscreen* atau penganturan kontras cahaya otomatis, yang tidak dimiliki oleh modul LCD biasa.
4. Mempercepat pengembangan produk: dengan menggunakan modul LCD yang sudah dilengkapi dengan IC2, pengembangan produk dapat mempercepat proses pengembangan, karena tidak perlu merancang sirkuit protocol tampilan dari awal.
5. Menghemat ruang: dalam sebuah sistem yang membutuhkan banyak komponen elektronik, penggunaan modul LCD dengan IC2 dapat menghemat ruang pada papan sirkuit karena tidak memerlukan sirkuit control tampilan yang besar dan rumit yang membuat hasil penampilan layar yang kurang maksimal.

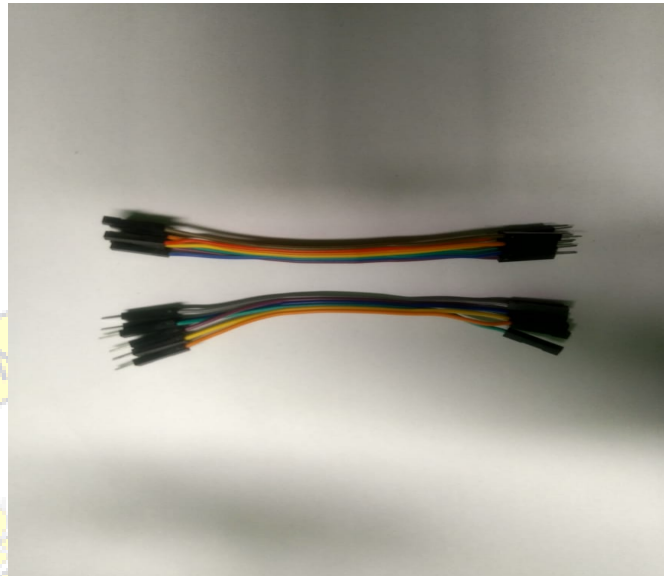


Gambar 2.3 LCD ic2
(sumber: dokumen pribadi)

d. Kabel jumper *male female*

Kabel atau dalam Bahasa Inggris disebut dengan *electrical cable* media untuk menghantarkan arus listrik yang terdiri dari konduktor dan isolator. Konduktor atau bahan penghantar listrik biasanya kabel menggunakan bahan tembaga dan bahan aluminium. *Electric cable* adalah media untuk menghantar arus listrik. Sedangkan isolator atau bahan yang sulit? Atau tidak menghantarkan arus listrik yang digunakan oleh kabel listrik yaitu bahan *thermoplastic* dan bahan *thermosetting* yakni *polymer* (*plastic* dan *ruber*) media pelapisan isolator pemanasan dan pendinginan kabel. Kabel ini merupakan sejumlah *wire* (kawat)

terisolator yang diikat bersama dan membentuk transmisi jalur multikonduktor.



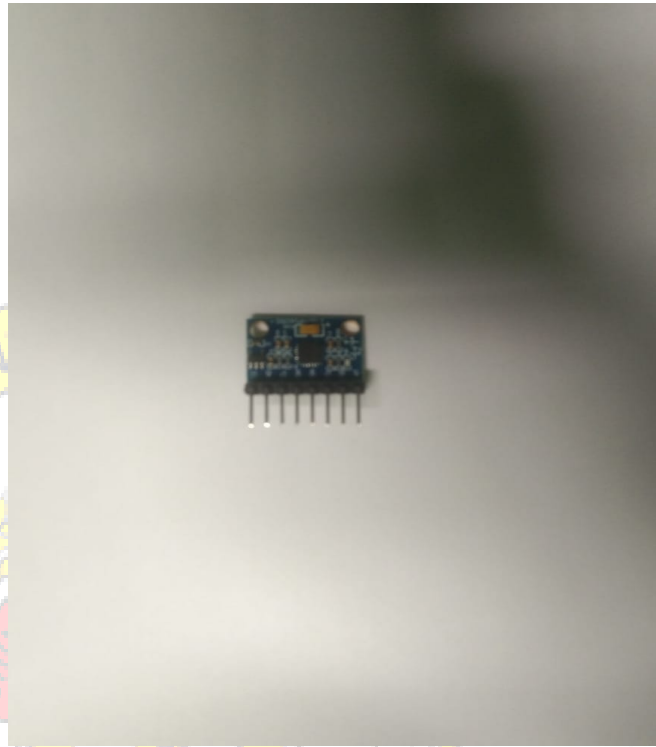
Gambar 2.4 kabel jumper
(sumber: dokumen pribadi)

e. MPU 3050 *gyroscope*

Sensor mpu 3050 merupakan gabungan dari sensor *accelerometer* dan *gyroscope* dimana dengan menggunakan kedua sensor ini, dapat menghasilkan output berupa orientasi. *Accelerometer* digunakan untuk mengetahui percepatan *gravitasi*, sedangkan *gyroscope* digunakan untuk mengetahui kecepatan sudut.

Fungsi dari *accelerometer* dan *gyroscope* akan menghasilkan orientasi (kemiringan terhadap sumbu X dan Y) begitu halnya dengan kemiringan kapal terhadap sumbu dengan sensor ini missalkan miring depan-belakang menggunakan sumbu X dan samping kanan-kiri menggunakan sumbu Y jika ingin

mengetahui sumbu keseluruhan mengetahui orientasi terhadap 3 sumbu yaitu: (X, Y, dan Z).



Gambar 2.5 MPU 6050
(sumber: dokumen pribadi)

f. *Sensor water level*

Water level merupakan sensor yang berfungsi untuk mendeteksi ketinggian air dengan *output* analog kemudian diolah menggunakan *mikrokontroler*.

Cara kerja sensor ini adalah pembacaan resistansi yang dihasilkan air yang mengenai garis atau pelampung pada bagian sensor yang mengenai arus listrik yang menimbulkan kontak pada pelampung sensor. Air dalam suatu wadah selalu meraih tekanan air agar sesor yang terbaca sesuai dengan harapan. Berikut ini adalah beberapa fungsi kegunaan sensor *water level*:

1. Pengendalian pompa air: sensor *water level* dapat digunakan untuk mengendalikan pompa air agar tidak terlalu berlebihan atau kekurangan air. Sensor ini dapat mendeteksi *level* air yang rendah atau tinggi, sehingga dapat mengaktifkan atau mematikan pompa air sesuai kebutuhan.
2. Pendekteksi banjir: sensor *water level* dapat digunakan untuk mendeteksi keberadaan air dalam ruangan atau area tertentu. Sensor ini dapat memberikan sinyal atau alarm jika terdeteksi level air yang melebihi batas tertentu, sehingga dapat mencegah terjadinya peluapan air yang berlebih.
3. Monitoring kapal: sensor *water level* juga dapat digunakan pada bagian kapal untuk memonitor level air didalam kapal. Sensor ini dapat membantu menghindari resiko kapal karam atau kandas karena air yang masuk ke dalam kapal tidak terkonrol dengan baik.



Gambar 2.6 sensor water level

(sumber: dokumen pribadi)

g. Pompa air

Pompa air adalah alat yang mampu menghasilkan aliran pompa menghasilkan aliran cairan menggunakan energi kinetic. Oleh karena itu memiliki beberapa elemen dasar yaitu:

1. Pintu masuk: tempat cairan diserap.
2. Motor + baling-baling: yang bertanggung jawab menghasilkan energi kinetic yang mengektaksi air dari saluran masuk dan mengirimkannya melalui saluran yang dituju atau keluar.
3. Keluar: ini adalah proses saluran masuk dimana cairan yang didorong oleh daya pompa air akan keluar, Pompa banyak digunakan dalam banyak keperluan dan perangkat sesuai kebutuhannya. Dari industry, hingga mesin penyalur air, sistem irigasi otomatis, sistem pasokan, pabrik pengolahan dan lainnya. Karena alasan ini banyak model dipasaran dengan daya dan kapasitas yang berbeda, Dari yang terkecil hingga terbesar.



Gambar 2.7 pompa air 6-12v

(sumber: dokumen pribadi)

h. Lampu indikator LED

LED atau disebut juga dengan *Light Emitting Diode* merupakan komponen elektronik yang berfungsi memancarkan cahaya monokromatik saat mendapatkan tegangan LED tergolong keluarga dioda yang terbuat dari bahan dasar semikonduktor.

LED lampu dapat menghasilkan warna cahaya yang tergantung dari jenis bahan semikonduktornya. LED juga bisa menghasilkan sinar inframerah yang tidak terlihat oleh mata seperti yang digunakan pada alat remote berupa remote AC, TV, dan masih banyak lagi. Lampu ini dipasang bersamaan bersama sensor yang akan digunakan sebagai pertanda alarm yang menyala pada lampu LED nantinya saat sensor bekerja.



Gambar 2.8 lampu LED 1,5v
(sumber: dokumen pribadi)

i. Saklar

Sekelar merupakan perangkat yang biasanya digunakan untuk memutus dan meneruskan arus listrik. Saklar pada mulanya adalah alat penyambung atau pemutus aliran listrik. Selain itu untuk jaringan arus listrik kuat atau lemah saklar berbentuk beraneka ragam mulai dari yang besar hingga terkecil tergantung spek dan kebutuhannya. Secara sederhana saklar terdiri dari dua logam yang dibilah dan bisang menyambung dan memutus sesuai dengan keadaan on atau off dalam penggunaannya. Biasanya bahan material yang dipilih yaitu yang tahan terhadap korosi. Logam yang digunakan terbuat dari bahan material oksida biasa, maka dari itu saklar akan sering malfungsi dan dampaknya saklar tidak dapat bekerja dengan semestinya. Untuk mengurangi hal seperti ini, bahan yang di gunakan harus dilapisi atau disepuh dengan anti karat. Saklar dengan model tombol atau *push button* diaplikasikan pada sensor mekanik karena alat ini biasa nya digunakan pada sensor mekanik kare alat ini bisa diaplikasikan pada mikrokontroler untuk simtem pengaturan control yang dilengkapi dengan sistem anti guncamg *bouncing* Saklar dibuat dalam banyak jenis dan kebutuhan ukuran yang bervariasi. Biasanya memiliki beberapa *set point* atau kontak yang dikendalikan oleh kenop atau *actuator* yang sama, kontak dapat bekerja bersamaan, berurutan atau bergantian. Saklar dapat dioperasikan secara manual dan otomatis contoh pada saklar manual yaitu: saklar lampu, tombol ketik laptop. Saklar dapat

berfungsi sebagai elemen pengendali pada mesin, ketinggian cairan, tekanan atau temperatur. Ada berbagai bentuk saklar khusus seperti saklar *toggle switch*, saklar putar, saklar merkuri, saklar tombol tekan, saklar pembalik, pemutus sirkuit atau relai. Saklar yang paling banyak digunakan untuk pengontrol pencahayaan beberapa saklar dapat dikonek ke sirkuit berdaya tinggi harus memiliki kriteria khusus untuk mencegah percikan api yang dampaknya akan merusak saklar dan komponen lainnya.

Saklar pada rancang bangun ini sangatlah penting dan berguna juga untuk memutuskan dan menyambungkan arus listrik. Peneliti menggunakan saklar model sederhana agar rancangan bisa bekerja dengan semestinya dengan desain yang sudah ditentukan. Saklar juga dapat digunakan untuk mengatur kecepatan motor listrik atau untuk menghilangkan arus listrik dari satu sirkuit ke sirkuit lainnya.



Gambar 2.9 saklar
(sumber: dokumen pribadi)

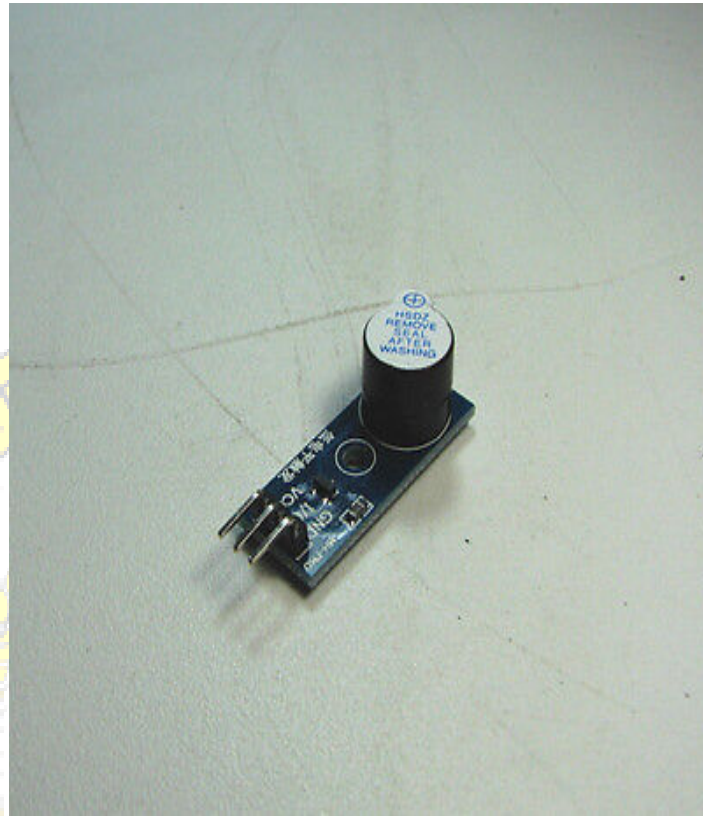
j. Alarm buzzer

Alarm buzzer adalah sebuah perangkat yang digunakan untuk memberikan peringatan atau sinyal suara yang kuat dan jelas. Buzzer ini sering digunakan dalam berbagai aplikasi seperti sistem keamanan, peralatan industry, permainan elektronik, atau perangkat lain yang memerlukan peringatan atau perhatian khusus.

Secara teknis, alarm buzzer terdiri dari sebuah suara, biasanya berupa elemen *piezoelektrik* atau speaker, dan sebuah sirkuit pengendali. Sirkuit pengendali tersebut menerima sinyal masukan dari sensor atau sistem control, kemudian mengaktifkan sumber suara untuk menghasilkan suara yang diinginkan. Buzzer dapat diatur untuk menghasilkan berbagai macam suara, mulai dari bunyi monoton seperti “bip” atau “bip-bip”, sehingga suara berkelajutan yang lebih kompleks. Buzzer juga dapat dikonfigurasi dengan pola bunyi tertentu, seperti berdenyut atau berkedip, untuk mengindikasikan jenis peringatan atau pesan yang berbeda. Dalam aplikasi keamanan,

Alarm buzzer sering digunakan untuk memberikan peringatan dini terhadap keadaan darurat, seperti kebakaran, instruksi, atau bahaya lainnya. Buzzer juga digunakan sebagai indicator kesalahan atau peringatan dalam peralatan elektronik atau industry, untuk memberi tahu pengguna atau operator tentang masalah yang perlu ditangani. Secara umum alarm buzzer adalah perangkat suara yang dirancang untuk membangunkan perhatian dan memberikan

peringatan melalui suara yang jelas dan terdengar oleh *crew* kapal yang berjaga.



Gambar 2.10 buzzer
(sumber: dokumen pribadi)

k. *Step Down 12V*

step down mengacu pada proses mengurangi tegangan listrik dari 12V menjadi tingkatan yang lebih rendah. Dalam konteks umum, "*step down*" berarti mengurangi tegangan yang lebih tinggi menjadi tegangan yang lebih rendah. Tegangan 12V umumnya digunakan dalam berbagai aplikasi, seperti sistem otomotif, perangkat elektronik *portable*, atau perangkat satu daya. Namun, ada situasi dimana diperlukan tegangan yang lebih rendah, misalnya, ketika

mengoperasikan komponen elektronika yang membutuhkan 5V atau 3.3V. dalam hal ini dapat menggunakan poerangkat seperti regulator tegangan *step-down* (*buck regulator*) atau mengubah tegangan DC-DC untuk menurunkan tengangan dari 12V menjadi tingkatan yang diinginkan. Pengaturan tengangan *input* yang lebih tinggi dalam menghasilkan tengangan *output* yang lebih rendah dengan efisiensi yang tinggi.

Pengaturan tengangan ini penting karena memungkinkan perangkat atau sistem yang membutuhkan tengangan yang lebih rendah dapat berpotensi dengan baik dan aman.

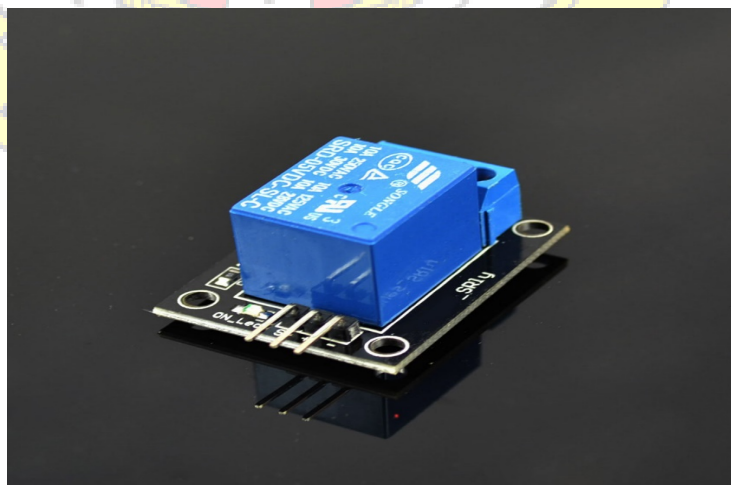


Gambar 2.11 komponen *step down*
(sumber: dokumen pribadi)

1. Relay

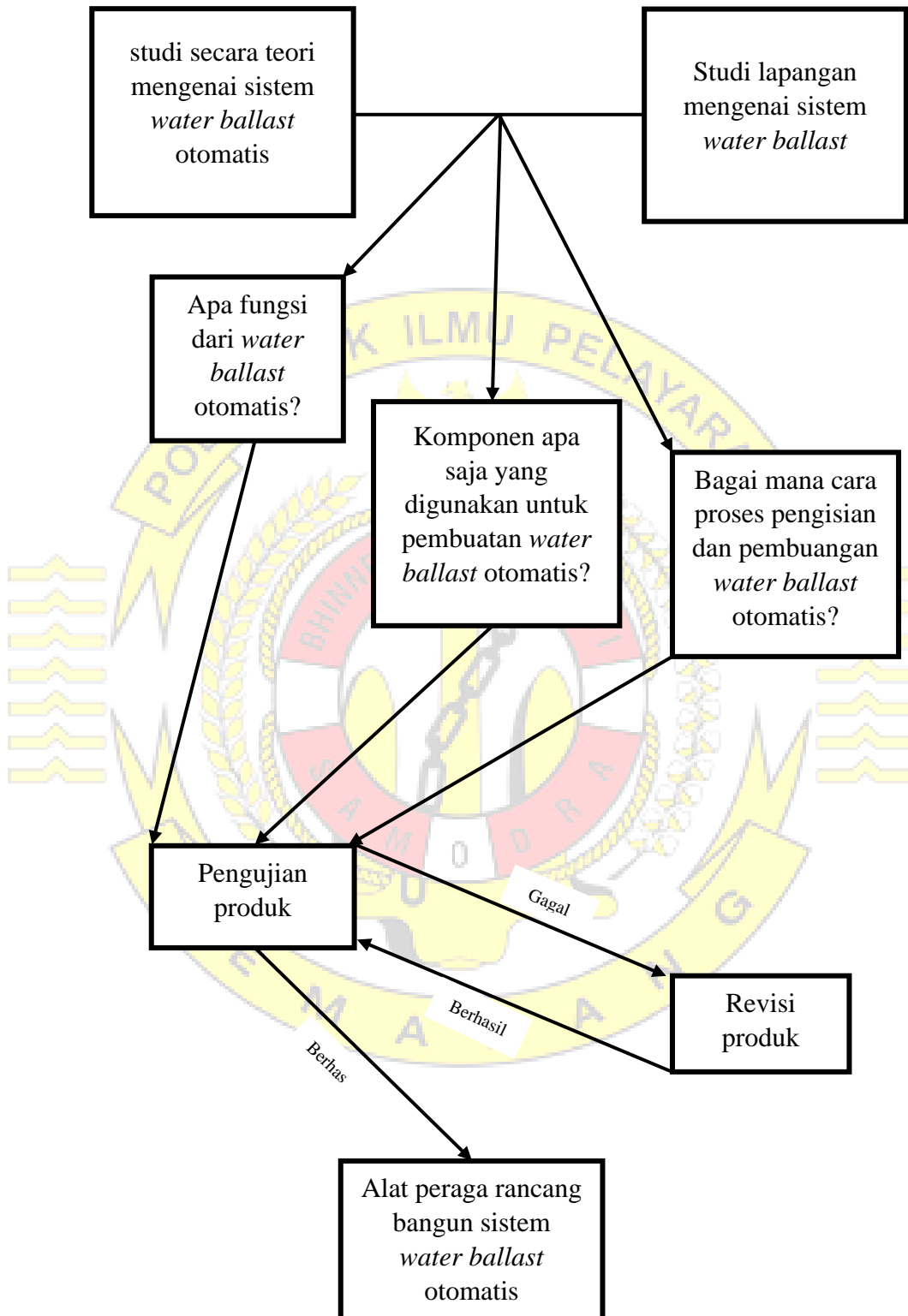
Relay terdiri dari beberapa bagian komponen bagian utama, termasuk kumparan elektromagnetik, kontak saklar (biasanya terdiri dari kontak normal terbuka dan normal tertutup), dan beberapa terminal koneksi. *Relay* digunakan dalam berbagai aplikasi, seperti sistem otomotif, kendali industri, peratan rumah tangga, sistem proteksi, dan banyak lagi. Fungsi mencakup pengendalian motor, pengalihan arus listrik, pengendalian lampu, pengontrol suhu, dan berbagai kegunaan lainnya.

Dengan menggunakan *relay*, sinyal control yang kecil dapat mengontrol perangkat atau sirkuit yang memerlukan daya atau tegangan yang lebih tinggi. Hal ini memungkinkan isolasi galvanic antara sinyal control dan beban listrik, serta memberikan kemampuan untuk mengintegrasikan dan mengontrol berbagai perangkat listrik dengan aman dan efektif.



Gambar 2.12 komponen *relay*
(sumber: dokumen pribadi)

A. Kerangka berfikir



Kerangka berfikir 2.1

B. Hipotesis

Hipotesis merupakan suatu rencana yang berisi tentang penjelasan dari semua aspek yang di jadikan sebagai bahan penelitian yang berdasarkan pada hasil penelitian tersebut, kerangka berfikir biasanya juga meliputi antara sebuah variable yang lainnya terdapat sebab, akibat atau lebih dari dua variable tersebut.

1. *Water ballast*

Ballast kapal adalah salah satu system pelayanan kapal yang mengangkut dan mengisi air *ballast*. Sistem pompa *ballast* di tujukan untuk menyesuaikan tingkat kemiringan dan *draft*, sebagai akibat dari perubahan muatan kapal sehingga stabilitas kapal dapat dipertahankan. Tangki *ballast* diisi dan dikosongkan dengan saluran pipa yang sama, jika stop *valve* di pasang pada system ini. Jumlah berat *ballast* yang di butuhkan untuk kapal rata-rata 10% sampai 20% dari displacement kapal.

2. Pengoprasian *ballast*

Cara kerja *ballast*, secara umum adalah untuk mengisi tangka *ballast* yang berada di *double bottom*, dengan air laut, yang diambil dari *seachas*. Melalui pompa *ballast*, dan saluran pipa utama dan cabang.

Fungsi *ballast* merupakan sistem untuk dapat memposisikan kapal dalam keadaan seimbang baik dalam keadaan trim depan maupun belakang, maupun keadaan oleng atau *rolling* kanan, kiri. Dalam perencanaannya adalah dengan memasukan air sebagai bahan *ballast* agar posisi kapal dapat

kembali pada posisi yang sempurna dan menambah gaya keseimbangan yang akan membuat kapal lebih stabil.

3. Pengembangan model

Pengembangan model adalah suatu usaha untuk menemukan penemuan, perbaikan atau pengembangan dan merupakan dasar untuk mengembangkan produk yang akan dihasilkan kaidah-kaidah dan metode ilmiah tertentu sehingga menghasilkan informasi yang di kehendaki. Pengembangan dapat dilakukan untuk kepentingan program, pelatihan, dan Pendidikan.

4. HMI (*Human Machine Interface*)

HMI merupakan sebuah sarana penghubung dan media komunikasi antara mesin dengan manusia, perangkat lunak antara mesin atau pabrik dan operator atau pengamat. Tujuan menggunakan HMI adalah untuk mengumpulkan dan menampilkan informasi dari proses pabrik. HMI dapat berupa pengendali dan visualisasi status baik dengan manual maupun *melalui visualisasi computer yang bersifat real time.*

BAB V

SIMPULAN DAN SARAN

A. Simpulan

Berdasarkan hasil penelitian yang telah dilakukan serta materi yang telah dibahas dan diuraikan pada karya tulis skripsi ini, maka dapat di simpulkan sebagai berikut:

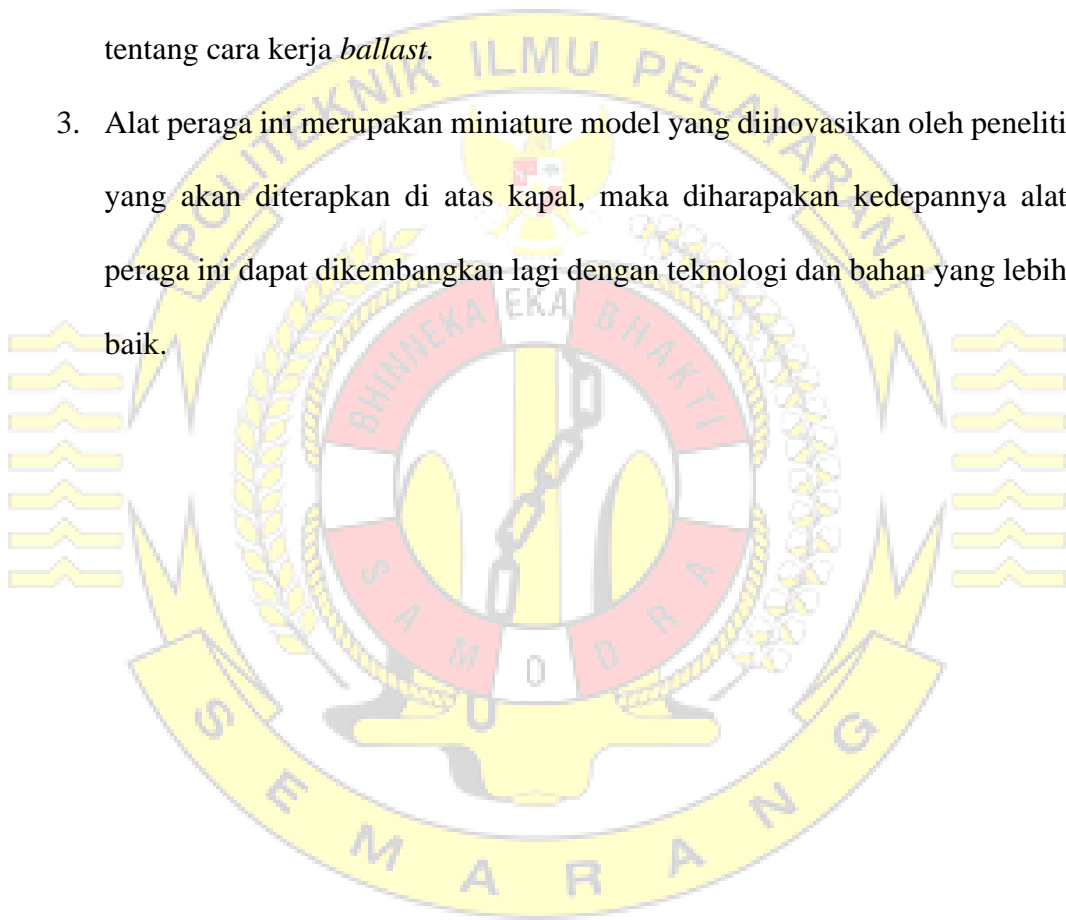
1. Apa fungsi dari *water ballast* otomatis untuk membantu mengatur dan menjaga stabilitas kapal selama berlayar, terutama saat kapal memiliki beban atau muatan yang berbeda dengan adanya sistem *water ballast* otomatis *crew* bisa mengerjakan tugas lain yang tidak ketergantungan oleh pengoprasian *ballast*.
2. Pada sarananya penelitian ini terpacu karena adanya inovasi yang tercipt dalam sebuah pemikiran peneliti, selain bahan yang mudah didapatkan, alat peraga ini juga tidak membutuhkan komponen-komponen yang rumit, serta mudah dalam mengoprasikan alat peraga ini. Alat peraga ini memiliki beberapa sensor yaitu: sensor water level yang berguna untuk memberikan sinyal dari kapasitas *ballast tank* yang akan digunakan nantinya, dikoneksikan dengan Arduino pada sebuah sensor yang akan digunakan, dan juga dilengkapi dengan sensor gyro untuk mendekteksi kemiringan yang ada pada sebuah kapal nantinya serta juga dilengkapi oleh panel instrument nominal yang ada pada informasi yang berjalan pada sensor, dan dilengkapi oleh lampu LED sebagai indikator yang akan menunjang sebuah informasi yang berjalan pada sistem yang akan dikerjakan.

3. Dalam proses perancangan system *water ballast* otomatis ini menggunakan beberapa sensor elektronik seperti RADUINO R3 sebagai main control utama dan akan memberikan perintah terhadap setiap sensor yang akan di kerjakan nantinya, yang bersangkutan dengan proses keseimbangan pada kapal dan dimana peran gyro sensor atau disebut dengan MPU 3050 sangat penting untuk kestabilan pada kapal. Motor yang berperan untuk memberikan masa terhadap kapal yang berupa air yang dimana mengisi di setiap tangki *ballast*. Untuk mengetahui volume maksimal terhadap tangki *ballast*, peneliti memasang sebuah alat yang bernama *SENSOR WATER LAVEL* kegunaan alat ini adalah sebagai pembatas maksimal volume air dan juga memberikan sinyal yang berupa suara atau di kenal sebagai alarm BUZER, karena sensor membaca volume air yang sudah mencapai titik maksimal, main sensor memerintahkan 1 POMPA *OVERBOARD* sebagai pompa air untuk membuang air yang berada di tangki *ballast* agar muatan dan *draft* sesuai. Agar prosesn pengisian *ballast* berjalan dengan baik ada 2 peran POMPA sebagai pengisi tangki *ballast* kanan maupun kiri, sebuah indicator yang berupa panel LCD dan dikolaborasikan dengan LED sebagai *second* indikator dengan adanya sebuah sensor untuk memerintahkan kinerja *auto* atau *manual*, untuk berjalan nya sebuah sensor *water ballast* otomatis.

B. Saran

Berdasarkan kesimpulan diatas yang telah dijabarkan, peneliti dapat menyampaikan saran dari pembuatan alat peraga pengoprasian *sistem water ballast otomatis* sebagai berikut:

1. Untuk pengembangan yang dapat dilakukan untuk penyempurnaan ini adalah dengan adanya penambahan pompa air yang digunakan untuk mengisi dan membuang *ballast* depan dan belakang sebagai media penambah berat dan stabilitas yang maksimal.
2. Untuk dosen/pengajar dapat memanfaatkan alat peraga *sistem water ballast otomatis* ini untuk media pembelajaran agar taruna lebih mudah memahami tentang cara kerja *ballast*.
3. Alat peraga ini merupakan miniature model yang diinovasikan oleh peneliti yang akan diterapkan di atas kapal, maka diharapkan kedepannya alat peraga ini dapat dikembangkan lagi dengan teknologi dan bahan yang lebih baik.



DAFTAR PUSTAKA

- Thomas. 1987. “ contoh sistem kontrol diatas kapal”
<https://ejurnal.its.ac.id/index.php/teknik/article/viewFile/3530/1492> pada 25 februari 2023.
- Jhson, F 2002. “sistem ballast pada kapal”
<https://www.intechopen.com/chapters/78698> pada 28 februari 2023
- Ardi, Rizal . 2011. “cara mengukur hambatan dengan multimeter”,
[https://belajar.com/cara - mengukur hambatan dengan multimeter/](https://belajar.com/cara-mengukur-hambatan-dengan-multimeter/),diakses pada 10 februari 2023.
- Reza. 2014. “fungsi dan jenis diode” [https://pinterelektro.com /ic- diode/](https://pinterelektro.com/ic-diode/),
Diakses pada 22 maret 2023.
- Khairi zaimudin. 2012. “efektifitas media maket sebagai presentasi karya perancangan arsitektur di era digital”,*skripsi*. Depok: universitas Indonesia.
- Megindo, adiel. 2016. Sistem *water control* menggunakan kendali volume air pada tangki ballast berbasis *ARDUINO UNO R3 volume 15*. Bandung: universitas teknologi bandung.
- Putra, nusa. 2015. *Research & development* penelitian dan pengembangan. Jakarta: PT. Raja grafindo persada.
- Sugiyono. 2015. *Metode penelitian kuantitatif, kualitatif dan R&D*. Surabaya PT. pelita .
- Technodand. 2016. “cara melakukan pengecekan pada komponen IC regulator”,
[https://www.technodand.net/2016/10/cara-melakukan-pengecekan – pada.](https://www.technodand.net/2016/10/cara-melakukan-pengecekan-pada-)
html, diakses pada 22 mei 2013.
- Handoyo. 2007. “metode penelitian kualitatif dalam mengerjakan skripsi”
<https://penerbitdepublish.com/metode-penelitian-kualitatif/> pada 23 maret 2023.

LAMPIRAN 1

KUESIONER

RANCANG BANGUN SISTEM *WATER BALLAST* OTOMATIS

Dalam rangka penyelesaian Skripsi. Saya, Faris selaku taruna dewasa tingkat akhir bermaksud melakukan penelitian ilmiah untuk penyusunan skripsi dengan judul “Rancang bangun sistem *water ballast* otomatis”. Sehubungan dengan hal tersebut saya sangat mengharapkan kesediaan bapak/ibu/saudara/i/taruna/i untuk meluangkan waktunya sejenak untuk mengisi beberapa pertanyaan pada kuesioner ini.

Atas perhatian dan kerja samanya, saya ucapkan terimakasih.

Nama :

Jenis kelamin :

Usia :

PETUNJUK PENGISIAN

1 = YA

0 = TIDAK

PERTANYAAN

NO	Pertanyaan	Ya	Tidak
1.	Apakah saudara mengetahui sistem <i>water ballast</i> .		
2.	Apakah saudara pernah mengoprasiakan pompa <i>ballast</i> .		

3.	Apakah saudara setuju atau tidak setuju dengan adanya sistem <i>water ballast otomatis</i> dikapal.		
4.	Apakah saudara pernah atau tidak pernah mengalami kerusakan pada sistem atau komponen <i>ballast</i> .		
5.	Apakah saudara mempelajari sistem <i>ballast</i> dikapal.		
6.	Apakah sistem <i>ballast</i> dikapal sangat penting.		
7.	Apakah saudara mengetahui fungsi dari <i>ballast</i> dikapal.		
8.	Apakah dikapal saudara menggunakan media air sebagai penyeimbang kapal.		
9.	Jika dikapal saudara menggunakan sistem <i>water ballast</i> otomatis apakah saudara akan menggunakannya setiap saat/ <i>stanby</i> .		
10.	Jika sistem <i>ballast</i> dikapal saudara bermasalah apakah saudara memperbaiki sendiri atau menunggu tenaga ahli dari darat.		

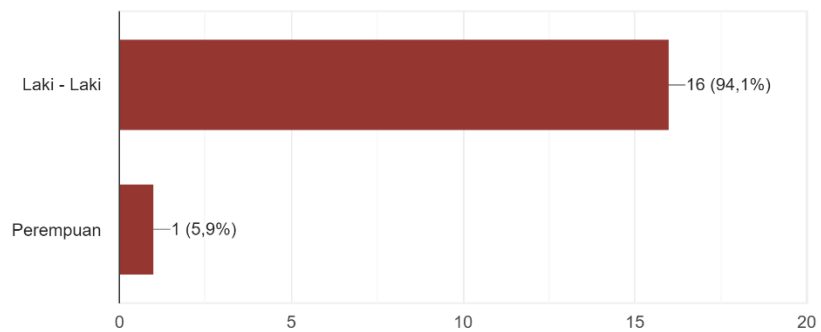
LAMPIRAN 2

BUKTI DATA

Bukti data *google form* mengenai pemahaman mengenai *sisyem water ballast otomatis*

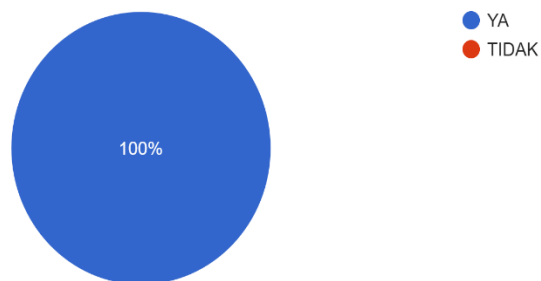
Jenis kelamin

17 jawaban



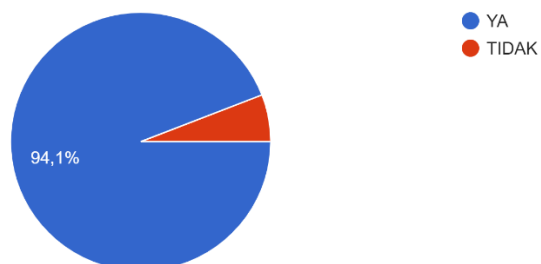
1. Apakah saudara mengetahui sistem water ballast.

17 jawaban

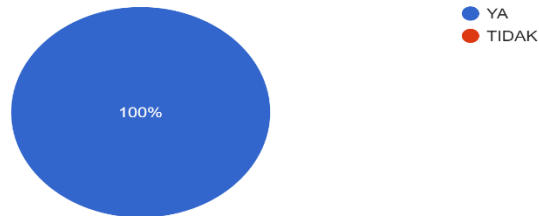


2. Apakah saudara pernah mengoprasiakan pompa ballast.

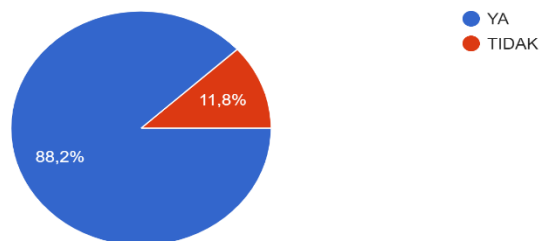
17 jawaban



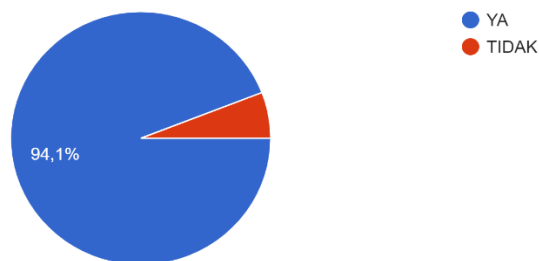
3. Apakah saudara setuju atau tidak setuju dengan adanya sistem water ballast otomatis dikapal.
17 jawaban



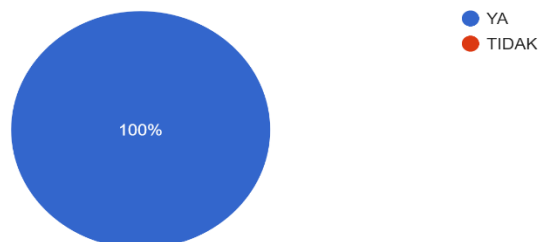
4. Apakah saudara pernah atau tidak pernah mengalami kerusakan pada sistem atau komponen ballast.
17 jawaban



5. Apakah saudara mempelajari sistem ballast dikapal.
17 jawaban

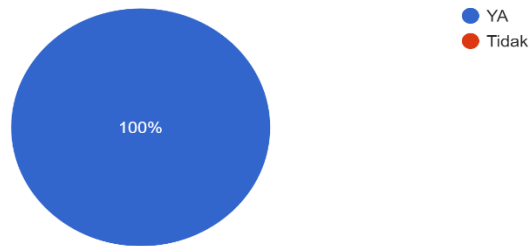


6. Apakah sistem ballast dikapal sangat penting.
17 jawaban



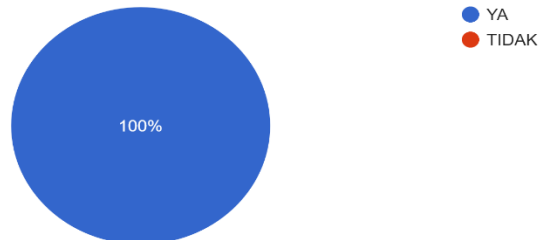
7. Apakah saudara mengetahui fungsi dari ballast dikapal.

17 jawaban



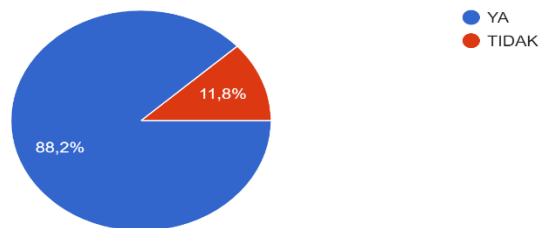
8. Apakah dikapal saudara menggunakan media air sebagai penyeimbang kapal.

17 jawaban



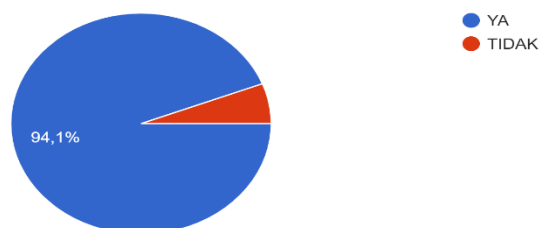
9. Jika dikapal saudara menggunakan sistem water ballast otomatis apakah saudara akan menggunakannya setiap saat/standby.

17 jawaban






10. Jika sistem ballast kapal saudara bermasalah apakah saudara memperbaiki sendiri atau menunggu tenaga ahli dari darat.

17 jawaban



LAMPIRAN 3

LEMBAR VALIDASI

	KEMENTERIAN PERHUBUNGAN				
	BADAN PENGEMBANGAN SDM PERHUBUNGAN				
	BADAN LAYANAN UMUM				
	POLITEKNIK ILMU PELAYARAN SEMARANG				
JALAN SINGOSARI 2A SEMARANG KODE POS 50242	TELP (62) 024-8311527 (62) 024-8311528	FAX : (62) 024-8311529	Email : info@pip-semarang.ac.id	ISO 9001:2015 Certificate No : 12C 90903	

SURAT KETERANGAN VALIDASI

Yang bertanda tangan dibawah ini:

Nama : Dr. ANDY WAHYU HERMANTO, S.T.,M.T

Jabatan : Dosen Pengampu Sistem Kelistrikan dan Kontrol

Instansi : PIP Semarang

Menyatakan bahwa instrument penelitian dengan judul:

“Rancang bangun *system water ballast otomatis* menggunakan Arduino R3”

Dari Taruna:

Nama : FARIZ FAUZHAN

Program Studi : D-IV TEKNIKA

NIT : 551811236931T

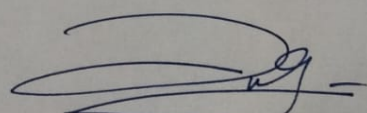
(~~Layak~~/~~Tidak Layak~~)*dipergunakan untuk siding skripsi dengan menambahkan saran sebagai berikut:

1. Penyempurnaan Body fisik dari kapal
2. Tambahkan keterangan terkait fungsi dr Tombol 3
3. Secara gansibesar sudah sangat bagus dan layak utk diujikan

Demikian surat keterangan ini kami buat untuk dapat dipergunakan sebagaimana mestinya.

Semarang, 12 Juli 2023

Validator


Dr. ANDY WAHYU HERMANTO ,S.T.,M.T

*) coret yang tidak perlu



**KEMENTERIAN PERHUBUNGAN
BADAN PENGEMBANGAN SDM PERHUBUNGAN
BADAN LAYANAN UMUM
POLITEKNIK ILMU PELAYARAN SEMARANG**

JALAN SINGOSARI 2A
SEMARANG
KODE POS 50242

TELP. (62) 024-8311527
(62) 024-8311528

FAX : (62) 024-8311529
Email : info@pip-semarang.ac.id
Home Page : www.pip-semarang.ac.id



SURAT KETERANGAN VALIDASI

Yang bertanda tangan dibawah ini:

Nama : Capt. SUROTO, M.Mar
Jabatan : Dosen Pengampu Sistem Kontruksi dan Stabilitas kapal
Instansi : PIP Semarang

Menyatakan bahwa instrument penelitian dengan judul:

"Rancang bangun *system water ballast otomatis* menggunakan Arduino R3"

Dari Taruna:

Nama : FARIZ FAUZIAN
Program Studi : D-IV TEKNIKA
NIT : 551811236931 T

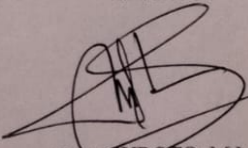
(~~Layak~~/~~Tidak Layak~~)* dipergunakan untuk siding skripsi dengan menambahkan saran sebagai berikut:

1. *Perlu disempurnakan ballast atau beban pada tanki
minyak di Gln muatan Gndah Fulldown ballast tidak beban.*
2. *Perlu ditambah tambak "Ship Particular"*

Demikian surat keterangan ini kami buat untuk dapat dipergunakan sebagaimana mestinya.

Semarang, 14 Juli 2023

Validator


 Capt. SUROTO, M.Mar

*) coret yang tidak perlu

LAMPIRAN 4

PROGRAM CODING SENSOR

TURBIDITY CODING ARDUINO SENSOR

```
// MotionApps 2.0 DMP implementation, built using the MPU-6050EVb evaluation board
#define MPU6050_INCLUDE_DMP_MOTIONAPPS20

#include "MPU6050_6Axis_MotionApps20.h"

// Tom Carpenter's conditional PROGMEM code
// http://forum.arduino.cc/index.php?topic=129407.0
#ifdef __AVR__
    #include <avr/pgmspace.h>
#elif defined(ESP32)
    #include <pgmspace.h>
#else
    // Teensy 3.0 library conditional PROGMEM code from Fariz Fauzian
    #ifndef __PGMSPACE_H_
        #define __PGMSPACE_H_ 1
        #include <inttypes.h>

        #define PROGMEM
        #define PGM_P const char *
        #define PSTR(str) (str)
        #define F(x) x

        typedef void prog_void;
        typedef char prog_char;
        typedef unsigned char prog_uchar;
        typedef int8_t prog_int8_t;
        typedef uint8_t prog_uint8_t;
        typedef int16_t prog_int16_t;
        typedef uint16_t prog_uint16_t;
```

```
typedef int32_t prog_int32_t;
typedef uint32_t prog_uint32_t;

#define strcpy_P(dest, src) strcpy((dest), (src))
#define strcat_P(dest, src) strcat((dest), (src))
#define strcmp_P(a, b) strcmp((a), (b))

#define pgm_read_byte(addr) (*(const unsigned char *)(addr))
#define pgm_read_word(addr) (*(const unsigned short *)(addr))
#define pgm_read_dword(addr) (*(const unsigned long *)(addr))
#define pgm_read_float(addr) (*(const float *)(addr))

#define pgm_read_byte_near(addr) pgm_read_byte(addr)
#define pgm_read_word_near(addr) pgm_read_word(addr)
#define pgm_read_dword_near(addr) pgm_read_dword(addr)
#define pgm_read_float_near(addr) pgm_read_float(addr)
#define pgm_read_byte_far(addr) pgm_read_byte(addr)
#define pgm_read_word_far(addr) pgm_read_word(addr)
#define pgm_read_dword_far(addr) pgm_read_dword(addr)
#define pgm_read_float_far(addr) pgm_read_float(addr)

#endif
#endif

/* Source is from the InvenSense MotionApps v2 demo code. Original source is
 * unavailable, unless you happen to be amazing at decompiling binary by
 * hand (in which case, please contact me, and I'm totally serious).
 *
 * Also, I'd like to offer many, many thanks to Noah Zerkin for all of the
 * DMP reverse-engineering he did to help make this bit of wizardry
 * possible.
 */

// NOTE! Enabling DEBUG adds about 3.3kB to the flash program size.
// Debug output is now working even on ATmega328P MCUs (e.g. Arduino Uno)
```

```

// after moving string constants to flash memory storage using the F()
// compiler macro (Arduino IDE 1.0+ required).

//#define DEBUG

#ifndef DEBUG

#define DEBUG_PRINT(x) Serial.print(x)
#define DEBUG_PRINTF(x, y) Serial.print(x, y)
#define DEBUG_PRINTLN(x) Serial.println(x)
#define DEBUG_PRINTLNF(x, y) Serial.println(x, y)

#else

#define DEBUG_PRINT(x)
#define DEBUG_PRINTF(x, y)
#define DEBUG_PRINTLN(x)
#define DEBUG_PRINTLNF(x, y)

#endif

#define MPU6050_DMP_CODE_SIZE    1929 // dmpMemory[]
#define MPU6050_DMP_CONFIG_SIZE  192  // dmpConfig[]
#define MPU6050_DMP_UPDATES_SIZE  47   // dmpUpdates[]

/*
=====
===== *

| Default MotionApps v2.0 42-byte FIFO packet structure:           |
|                                                                     |
| [QUAT W][  ][QUAT X][  ][QUAT Y][  ][QUAT Z][  ][GYRO X][  ][GYRO Y][  ] |
| 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 |
|                                                                     |
| [GYRO Z][  ][ACC X][ ][ACC Y][ ][ACC Z][ ][  ] |
| 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41           |
|                                                                     |
*
=====
===== */

// this block of memory gets written to the MPU on start-up, and it seems

```


0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x09, 0x23, 0xA1, 0x35, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x03, 0x3F, 0x68, 0xB6, 0x79, 0x35, 0x28, 0xBC, 0xC6, 0x7E, 0xD1, 0x6C,
 0x80, 0x00, 0xFF, 0xFF, 0x40, 0x00, 0x00, 0x00, 0x00, 0x00, 0xB2, 0x6A, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x3F, 0xF0, 0x00, 0x00, 0x00, 0x30,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x25, 0x4D, 0x00, 0x2F, 0x70, 0x6D, 0x00, 0x00, 0x05, 0xAE, 0x00, 0x0C, 0x02, 0xD0,
 /* bank # 2 */

0x00, 0x00, 0x00, 0x00, 0x00, 0x65, 0x00, 0x54, 0xFF, 0xEF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x01, 0x00, 0x00, 0x44, 0x00, 0x01, 0x00, 0x05, 0x8B, 0xC1, 0x00, 0x00, 0x01, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x65, 0x00, 0x00, 0x00, 0x54, 0x00, 0x00, 0xFF, 0xEF, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x02, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x1B, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x1B, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 /* bank # 3 */

0xD8, 0xDC, 0xBA, 0xA2, 0xF1, 0xDE, 0xB2, 0xB8, 0xB4, 0xA8, 0x81, 0x91, 0xF7, 0x4A, 0x90,
 0x7F,
 0x91, 0x6A, 0xF3, 0xF9, 0xDB, 0xA8, 0xF9, 0xB0, 0xBA, 0xA0, 0x80, 0xF2, 0xCE, 0x81, 0xF3,
 0xC2,
 0xF1, 0xC1, 0xF2, 0xC3, 0xF3, 0xCC, 0xA2, 0xB2, 0x80, 0xF1, 0xC6, 0xD8, 0x80, 0xBA, 0xA7,
 0xDF,
 0xDF, 0xDF, 0xF2, 0xA7, 0xC3, 0xCB, 0xC5, 0xB6, 0xF0, 0x87, 0xA2, 0x94, 0x24, 0x48, 0x70,
 0x3C,
 0x95, 0x40, 0x68, 0x34, 0x58, 0x9B, 0x78, 0xA2, 0xF1, 0x83, 0x92, 0x2D, 0x55, 0x7D, 0xD8,
 0xB1,

0xB4, 0xB8, 0xA1, 0xD0, 0x91, 0x80, 0xF2, 0x70, 0xF3, 0x70, 0xF2, 0x7C, 0x80, 0xA8, 0xF1,
 0x01,
 0xB0, 0x98, 0x87, 0xD9, 0x43, 0xD8, 0x86, 0xC9, 0x88, 0xBA, 0xA1, 0xF2, 0x0E, 0xB8, 0x97,
 0x80,
 0xF1, 0xA9, 0xDF, 0xDF, 0xDF, 0xAA, 0xDF, 0xDF, 0xDF, 0xF2, 0xAA, 0x4C, 0xCD, 0x6C,
 0xA9, 0x0C,
 0xC9, 0x2C, 0x97, 0x97, 0x97, 0x97, 0xF1, 0xA9, 0x89, 0x26, 0x46, 0x66, 0xB0, 0xB4, 0xBA,
 0x80,
 0xAC, 0xDE, 0xF2, 0xCA, 0xF1, 0xB2, 0x8C, 0x02, 0xA9, 0xB6, 0x98, 0x00, 0x89, 0x0E, 0x16,
 0x1E,
 0xB8, 0xA9, 0xB4, 0x99, 0x2C, 0x54, 0x7C, 0xB0, 0x8A, 0xA8, 0x96, 0x36, 0x56, 0x76, 0xF1,
 0xB9,
 0xAF, 0xB4, 0xB0, 0x83, 0xC0, 0xB8, 0xA8, 0x97, 0x11, 0xB1, 0x8F, 0x98, 0xB9, 0xAF, 0xF0,
 0x24,
 0x08, 0x44, 0x10, 0x64, 0x18, 0xF1, 0xA3, 0x29, 0x55, 0x7D, 0xAF, 0x83, 0xB5, 0x93, 0xAF,
 0xF0,
 0x00, 0x28, 0x50, 0xF1, 0xA3, 0x86, 0x9F, 0x61, 0xA6, 0xDA, 0xDE, 0xDF, 0xD9, 0xFA, 0xA3,
 0x86,
 0x96, 0xDB, 0x31, 0xA6, 0xD9, 0xF8, 0xDF, 0xBA, 0xA6, 0x8F, 0xC2, 0xC5, 0xC7, 0xB2, 0x8C,
 0xC1,
 0xB8, 0xA2, 0xDF, 0xDF, 0xDF, 0xA3, 0xDF, 0xDF, 0xDF, 0xD8, 0xD8, 0xF1, 0xB8, 0xA8, 0xB2,
 0x86,
 /* bank # 4 */
 0xB4, 0x98, 0x0D, 0x35, 0x5D, 0xB8, 0xAA, 0x98, 0xB0, 0x87, 0x2D, 0x35, 0x3D, 0xB2, 0xB6,
 0xBA,
 0xAF, 0x8C, 0x96, 0x19, 0x8F, 0x9F, 0xA7, 0x0E, 0x16, 0x1E, 0xB4, 0x9A, 0xB8, 0xAA, 0x87,
 0x2C,
 0x54, 0x7C, 0xB9, 0xA3, 0xDE, 0xDF, 0xDF, 0xA3, 0xB1, 0x80, 0xF2, 0xC4, 0xCD, 0xC9, 0xF1,
 0xB8,
 0xA9, 0xB4, 0x99, 0x83, 0x0D, 0x35, 0x5D, 0x89, 0xB9, 0xA3, 0x2D, 0x55, 0x7D, 0xB5, 0x93,
 0xA3,
 0x0E, 0x16, 0x1E, 0xA9, 0x2C, 0x54, 0x7C, 0xB8, 0xB4, 0xB0, 0xF1, 0x97, 0x83, 0xA8, 0x11,
 0x84,
 0xA5, 0x09, 0x98, 0xA3, 0x83, 0xF0, 0xDA, 0x24, 0x08, 0x44, 0x10, 0x64, 0x18, 0xD8, 0xF1,
 0xA5,
 0x29, 0x55, 0x7D, 0xA5, 0x85, 0x95, 0x02, 0x1A, 0x2E, 0x3A, 0x56, 0x5A, 0x40, 0x48, 0xF9,
 0xF3,
 0xA3, 0xD9, 0xF8, 0xF0, 0x98, 0x83, 0x24, 0x08, 0x44, 0x10, 0x64, 0x18, 0x97, 0x82, 0xA8, 0xF1,
 0x11, 0xF0, 0x98, 0xA2, 0x24, 0x08, 0x44, 0x10, 0x64, 0x18, 0xDA, 0xF3, 0xDE, 0xD8, 0x83,
 0xA5,
 0x94, 0x01, 0xD9, 0xA3, 0x02, 0xF1, 0xA2, 0xC3, 0xC5, 0xC7, 0xD8, 0xF1, 0x84, 0x92, 0xA2,
 0x4D,

0xDA, 0x2A, 0xD8, 0x48, 0x69, 0xD9, 0x2A, 0xD8, 0x68, 0x55, 0xDA, 0x32, 0xD8, 0x50, 0x71,
 0xD9,
 0x32, 0xD8, 0x70, 0x5D, 0xDA, 0x3A, 0xD8, 0x58, 0x79, 0xD9, 0x3A, 0xD8, 0x78, 0x93, 0xA3,
 0x4D,
 0xDA, 0x2A, 0xD8, 0x48, 0x69, 0xD9, 0x2A, 0xD8, 0x68, 0x55, 0xDA, 0x32, 0xD8, 0x50, 0x71,
 0xD9,
 0x32, 0xD8, 0x70, 0x5D, 0xDA, 0x3A, 0xD8, 0x58, 0x79, 0xD9, 0x3A, 0xD8, 0x78, 0xA8, 0x8A,
 0x9A,
 0xF0, 0x28, 0x50, 0x78, 0x9E, 0xF3, 0x88, 0x18, 0xF1, 0x9F, 0x1D, 0x98, 0xA8, 0xD9, 0x08, 0xD8,
 0xC8, 0x9F, 0x12, 0x9E, 0xF3, 0x15, 0xA8, 0xDA, 0x12, 0x10, 0xD8, 0xF1, 0xAF, 0xC8, 0x97,
 0x87,
 /* bank # 5 */
 0x34, 0xB5, 0xB9, 0x94, 0xA4, 0x21, 0xF3, 0xD9, 0x22, 0xD8, 0xF2, 0x2D, 0xF3, 0xD9, 0x2A,
 0xD8,
 0xF2, 0x35, 0xF3, 0xD9, 0x32, 0xD8, 0x81, 0xA4, 0x60, 0x60, 0x61, 0xD9, 0x61, 0xD8, 0x6C,
 0x68,
 0x69, 0xD9, 0x69, 0xD8, 0x74, 0x70, 0x71, 0xD9, 0x71, 0xD8, 0xB1, 0xA3, 0x84, 0x19, 0x3D,
 0x5D,
 0xA3, 0x83, 0x1A, 0x3E, 0x5E, 0x93, 0x10, 0x30, 0x81, 0x10, 0x11, 0xB8, 0xB0, 0xAF, 0x8F,
 0x94,
 0xF2, 0xDA, 0x3E, 0xD8, 0xB4, 0x9A, 0xA8, 0x87, 0x29, 0xDA, 0xF8, 0xD8, 0x87, 0x9A, 0x35,
 0xDA,
 0xF8, 0xD8, 0x87, 0x9A, 0x3D, 0xDA, 0xF8, 0xD8, 0xB1, 0xB9, 0xA4, 0x98, 0x85, 0x02, 0x2E,
 0x56,
 0xA5, 0x81, 0x00, 0x0C, 0x14, 0xA3, 0x97, 0xB0, 0x8A, 0xF1, 0x2D, 0xD9, 0x28, 0xD8, 0x4D,
 0xD9,
 0x48, 0xD8, 0x6D, 0xD9, 0x68, 0xD8, 0xB1, 0x84, 0x0D, 0xDA, 0x0E, 0xD8, 0xA3, 0x29, 0x83,
 0xDA,
 0x2C, 0x0E, 0xD8, 0xA3, 0x84, 0x49, 0x83, 0xDA, 0x2C, 0x4C, 0x0E, 0xD8, 0xB8, 0xB0, 0xA8,
 0x8A,
 0x9A, 0xF5, 0x20, 0xAA, 0xDA, 0xDF, 0xD8, 0xA8, 0x40, 0xAA, 0xD0, 0xDA, 0xDE, 0xD8,
 0xA8, 0x60,
 0xAA, 0xDA, 0xD0, 0xDF, 0xD8, 0xF1, 0x97, 0x86, 0xA8, 0x31, 0x9B, 0x06, 0x99, 0x07, 0xAB,
 0x97,
 0x28, 0x88, 0x9B, 0xF0, 0x0C, 0x20, 0x14, 0x40, 0xB8, 0xB0, 0xB4, 0xA8, 0x8C, 0x9C, 0xF0,
 0x04,
 0x28, 0x51, 0x79, 0x1D, 0x30, 0x14, 0x38, 0xB2, 0x82, 0xAB, 0xD0, 0x98, 0x2C, 0x50, 0x50, 0x78,
 0x78, 0x9B, 0xF1, 0x1A, 0xB0, 0xF0, 0x8A, 0x9C, 0xA8, 0x29, 0x51, 0x79, 0x8B, 0x29, 0x51,
 0x79,
 0x8A, 0x24, 0x70, 0x59, 0x8B, 0x20, 0x58, 0x71, 0x8A, 0x44, 0x69, 0x38, 0x8B, 0x39, 0x40, 0x68,
 0x8A, 0x64, 0x48, 0x31, 0x8B, 0x30, 0x49, 0x60, 0xA5, 0x88, 0x20, 0x09, 0x71, 0x58, 0x44, 0x68,

/* bank # 6 */

0x11, 0x39, 0x64, 0x49, 0x30, 0x19, 0xF1, 0xAC, 0x00, 0x2C, 0x54, 0x7C, 0xF0, 0x8C, 0xA8,
0x04,
0x28, 0x50, 0x78, 0xF1, 0x88, 0x97, 0x26, 0xA8, 0x59, 0x98, 0xAC, 0x8C, 0x02, 0x26, 0x46, 0x66,
0xF0, 0x89, 0x9C, 0xA8, 0x29, 0x51, 0x79, 0x24, 0x70, 0x59, 0x44, 0x69, 0x38, 0x64, 0x48, 0x31,
0xA9, 0x88, 0x09, 0x20, 0x59, 0x70, 0xAB, 0x11, 0x38, 0x40, 0x69, 0xA8, 0x19, 0x31, 0x48, 0x60,
0x8C, 0xA8, 0x3C, 0x41, 0x5C, 0x20, 0x7C, 0x00, 0xF1, 0x87, 0x98, 0x19, 0x86, 0xA8, 0x6E,
0x76,
0x7E, 0xA9, 0x99, 0x88, 0x2D, 0x55, 0x7D, 0x9E, 0xB9, 0xA3, 0x8A, 0x22, 0x8A, 0x6E, 0x8A,
0x56,
0x8A, 0x5E, 0x9F, 0xB1, 0x83, 0x06, 0x26, 0x46, 0x66, 0x0E, 0x2E, 0x4E, 0x6E, 0x9D, 0xB8,
0xAD,
0x00, 0x2C, 0x54, 0x7C, 0xF2, 0xB1, 0x8C, 0xB4, 0x99, 0xB9, 0xA3, 0x2D, 0x55, 0x7D, 0x81,
0x91,
0xAC, 0x38, 0xAD, 0x3A, 0xB5, 0x83, 0x91, 0xAC, 0x2D, 0xD9, 0x28, 0xD8, 0x4D, 0xD9, 0x48,
0xD8,
0x6D, 0xD9, 0x68, 0xD8, 0x8C, 0x9D, 0xAE, 0x29, 0xD9, 0x04, 0xAE, 0xD8, 0x51, 0xD9, 0x04,
0xAE,
0xD8, 0x79, 0xD9, 0x04, 0xD8, 0x81, 0xF3, 0x9D, 0xAD, 0x00, 0x8D, 0xAE, 0x19, 0x81, 0xAD,
0xD9,
0x01, 0xD8, 0xF2, 0xAE, 0xDA, 0x26, 0xD8, 0x8E, 0x91, 0x29, 0x83, 0xA7, 0xD9, 0xAD, 0xAD,
0xAD,
0xAD, 0xF3, 0x2A, 0xD8, 0xD8, 0xF1, 0xB0, 0xAC, 0x89, 0x91, 0x3E, 0x5E, 0x76, 0xF3, 0xAC,
0x2E,
0x2E, 0xF1, 0xB1, 0x8C, 0x5A, 0x9C, 0xAC, 0x2C, 0x28, 0x28, 0x28, 0x9C, 0xAC, 0x30, 0x18,
0xA8,
0x98, 0x81, 0x28, 0x34, 0x3C, 0x97, 0x24, 0xA7, 0x28, 0x34, 0x3C, 0x9C, 0x24, 0xF2, 0xB0, 0x89,
0xAC, 0x91, 0x2C, 0x4C, 0x6C, 0x8A, 0x9B, 0x2D, 0xD9, 0xD8, 0xD8, 0x51, 0xD9, 0xD8, 0xD8,
0x79,

/* bank # 7 */

0xD9, 0xD8, 0xD8, 0xF1, 0x9E, 0x88, 0xA3, 0x31, 0xDA, 0xD8, 0xD8, 0x91, 0x2D, 0xD9, 0x28,
0xD8,
0x4D, 0xD9, 0x48, 0xD8, 0x6D, 0xD9, 0x68, 0xD8, 0xB1, 0x83, 0x93, 0x35, 0x3D, 0x80, 0x25,
0xDA,
0xD8, 0xD8, 0x85, 0x69, 0xDA, 0xD8, 0xD8, 0xB4, 0x93, 0x81, 0xA3, 0x28, 0x34, 0x3C, 0xF3,
0xAB,
0x8B, 0xF8, 0xA3, 0x91, 0xB6, 0x09, 0xB4, 0xD9, 0xAB, 0xDE, 0xFA, 0xB0, 0x87, 0x9C, 0xB9,
0xA3,
0xDD, 0xF1, 0x20, 0x28, 0x30, 0x38, 0x9A, 0xF1, 0x28, 0x30, 0x38, 0x9D, 0xF1, 0xA3, 0xA3,
0xA3,

```

    0xA3, 0xF2, 0xA3, 0xB4, 0x90, 0x80, 0xF2, 0xA3, 0xA3, 0xA3, 0xA3, 0xA3, 0xA3, 0xA3, 0xA3,
0xA3,
    0xA3, 0xB2, 0xA3, 0xA3, 0xA3, 0xA3, 0xA3, 0xA3, 0xB0, 0x87, 0xB5, 0x99, 0xF1, 0x28, 0x30,
0x38,
    0x98, 0xF1, 0xA3, 0xA3, 0xA3, 0xA3, 0x97, 0xA3, 0xA3, 0xA3, 0xA3, 0xF3, 0x9B, 0xA3, 0x30,
0xDC,
    0xB9, 0xA7, 0xF1, 0x26, 0x26, 0x26, 0xFE, 0xD8, 0xFF,

```

```
};
```

```
#ifndef MPU6050_DMP_FIFO_RATE_DIVISOR
```

```
#define MPU6050_DMP_FIFO_RATE_DIVISOR 0x01 // The New instance of the Firmware has this as the
default
```

```
#endif
```

```
// I Simplified this:
```

```
uint8_t MPU6050_6Axis_MotionApps20::dmpInitialize() {
    // reset device
    DEBUG_PRINTLN(F("\n\nResetting MPU6050..."));
    reset();
    delay(30); // wait after reset

    // enable sleep mode and wake cycle
    /*Serial.println(F("Enabling sleep mode..."));
    setSleepEnabled(true);
    Serial.println(F("Enabling wake cycle..."));
    setWakeCycleEnabled(true);*/

    // disable sleep mode
    setSleepEnabled(false);

    // get MPU hardware revision
    setMemoryBank(0x10, true, true);
    setMemoryStartAddress(0x06);
    DEBUG_PRINTLN(F("Checking hardware revision..."));

```

```

DEBUG_PRINT(F("Revision @ user[16][6] = "));
DEBUG_PRINTLN(readMemoryByte());
DEBUG_PRINTLN(F("Resetting memory bank selection to 0..."));
setMemoryBank(0, false, false);

// check OTP bank valid
DEBUG_PRINTLN(F("Reading OTP bank valid flag..."));
DEBUG_PRINT(F("OTP bank is "));
DEBUG_PRINTLN(getOTPBANKValid() ? F("valid!") : F("invalid!"));

// setup weird slave stuff (?)
DEBUG_PRINTLN(F("Setting slave 0 address to 0x7F..."));
setSlaveAddress(0, 0x7F);
DEBUG_PRINTLN(F("Disabling I2C Master mode..."));
setI2CMasterModeEnabled(false);
DEBUG_PRINTLN(F("Setting slave 0 address to 0x68 (self...)"));
setSlaveAddress(0, 0x68);
DEBUG_PRINTLN(F("Resetting I2C Master control..."));
resetI2CMaster();

delay(20);
DEBUG_PRINTLN(F("Setting clock source to Z Gyro..."));
setClockSource(MPU6050_CLOCK_PLL_ZGYRO);

DEBUG_PRINTLN(F("Setting DMP and FIFO_OFLOW interrupts enabled..."));
setIntEnabled(1<<MPU6050_INTERRUPT_FIFO_OFLOW_BIT|1<<MPU6050_INTERRUPT_D
MP_INT_BIT);

DEBUG_PRINTLN(F("Setting sample rate to 200Hz..."));
setRate(4); // 1khz / (1 + 4) = 200 Hz

DEBUG_PRINTLN(F("Setting external frame sync to TEMP_OUT_L[0]..."));
setExternalFrameSync(MPU6050_EXT_SYNC_TEMP_OUT_L);

DEBUG_PRINTLN(F("Setting DLPF bandwidth to 42Hz..."));

```

```

setDLPFMode(MPU6050_DLPF_BW_42);

DEBUG_PRINTLN(F("Setting gyro sensitivity to +/- 2000 deg/sec..."));
setFullScaleGyroRange(MPU6050_GYRO_FS_2000);

// load DMP code into memory banks
DEBUG_PRINT(F("Writing DMP code to MPU memory banks ("));
DEBUG_PRINT(MPU6050_DMP_CODE_SIZE);
DEBUG_PRINTLN(F(" bytes)"));
if (!writeProgMemoryBlock(dmpMemory, MPU6050_DMP_CODE_SIZE)) return 1; // Failed
DEBUG_PRINTLN(F("Success! DMP code written and verified.));

// Set the FIFO Rate Divisor int the DMP Firmware Memory
unsigned char dmpUpdate[] = {0x00, MPU6050_DMP_FIFO_RATE_DIVISOR};

writeMemoryBlock(dmpUpdate, 0x02, 0x02, 0x16); // Lets write the dmpUpdate data to the
Firmware image, we have 2 bytes to write in bank 0x02 with the Offset 0x16

//write start address MSB into register
setDMPConfig1(0x03);

//write start address LSB into register
setDMPConfig2(0x00);

DEBUG_PRINTLN(F("Clearing OTP Bank flag..."));
setOTPBANKValid(false);

DEBUG_PRINTLN(F("Setting motion detection threshold to 2..."));
setMotionDetectionThreshold(2);

DEBUG_PRINTLN(F("Setting zero-motion detection threshold to 156..."));
setZeroMotionDetectionThreshold(156);

DEBUG_PRINTLN(F("Setting motion detection duration to 80..."));
setMotionDetectionDuration(80);

```

```

    DEBUG_PRINTLN(F("Setting zero-motion detection duration to 0..."));
    setZeroMotionDetectionDuration(0);
    DEBUG_PRINTLN(F("Enabling FIFO..."));
    setFIFOEnabled(true);

    DEBUG_PRINTLN(F("Resetting DMP..."));
    resetDMP();

    DEBUG_PRINTLN(F("DMP is good to go! Finally.));

    DEBUG_PRINTLN(F("Disabling DMP (you turn it on later)..."));
    setDMPEnabled(false);

    DEBUG_PRINTLN(F("Setting up internal 42-byte (default) DMP packet buffer..."));
    dmpPacketSize = 42;

    DEBUG_PRINTLN(F("Resetting FIFO and clearing INT status one last time..."));
    resetFIFO();
    getIntStatus();

    return 0; // success
}

// Nothing else changed

bool MPU6050_6Axis_MotionApps20::dmpPacketAvailable() {
    return getFIFOCount() >= dmpGetFIFOPacketSize();
}

// uint8_t MPU6050_6Axis_MotionApps20::dmpSetFIFORate(uint8_t fifoRate);
// uint8_t MPU6050_6Axis_MotionApps20::dmpGetFIFORate();
// uint8_t MPU6050_6Axis_MotionApps20::dmpGetSampleStepSizeMS();
// uint8_t MPU6050_6Axis_MotionApps20::dmpGetSampleFrequency();
// int32_t MPU6050_6Axis_MotionApps20::dmpDecodeTemperature(int8_t tempReg);

```

```

//uint8_t MPU6050_6Axis_MotionApps20::dmpRegisterFIFORateProcess(inv_obj_func func, int16_t
priority);

//uint8_t MPU6050_6Axis_MotionApps20::dmpUnregisterFIFORateProcess(inv_obj_func func);

//uint8_t MPU6050_6Axis_MotionApps20::dmpRunFIFORateProcesses();

// uint8_t MPU6050_6Axis_MotionApps20::dmpSendQuaternion(uint_fast16_t accuracy);

// uint8_t MPU6050_6Axis_MotionApps20::dmpSendGyro(uint_fast16_t elements, uint_fast16_t accuracy);

// uint8_t MPU6050_6Axis_MotionApps20::dmpSendAccel(uint_fast16_t elements, uint_fast16_t accuracy);

// uint8_t MPU6050_6Axis_MotionApps20::dmpSendLinearAccel(uint_fast16_t elements, uint_fast16_t
accuracy);

// uint8_t MPU6050_6Axis_MotionApps20::dmpSendLinearAccelInWorld(uint_fast16_t elements,
uint_fast16_t accuracy);

// uint8_t MPU6050_6Axis_MotionApps20::dmpSendControlData(uint_fast16_t elements, uint_fast16_t
accuracy);

// uint8_t MPU6050_6Axis_MotionApps20::dmpSendSensorData(uint_fast16_t elements, uint_fast16_t
accuracy);

// uint8_t MPU6050_6Axis_MotionApps20::dmpSendExternalSensorData(uint_fast16_t elements,
uint_fast16_t accuracy);

// uint8_t MPU6050_6Axis_MotionApps20::dmpSendGravity(uint_fast16_t elements, uint_fast16_t
accuracy);

// uint8_t MPU6050_6Axis_MotionApps20::dmpSendPacketNumber(uint_fast16_t accuracy);

// uint8_t MPU6050_6Axis_MotionApps20::dmpSendQuantizedAccel(uint_fast16_t elements, uint_fast16_t
accuracy);

// uint8_t MPU6050_6Axis_MotionApps20::dmpSendEIS(uint_fast16_t elements, uint_fast16_t accuracy);

uint8_t MPU6050_6Axis_MotionApps20::dmpGetAccel(int32_t *data, const uint8_t* packet) {
    // TODO: accommodate different arrangements of sent data (ONLY default supported now)

    if (packet == 0) packet = dmpPacketBuffer;

    data[0] = (((uint32_t)packet[28] << 24) | ((uint32_t)packet[29] << 16) | ((uint32_t)packet[30] << 8) |
packet[31]);

    data[1] = (((uint32_t)packet[32] << 24) | ((uint32_t)packet[33] << 16) | ((uint32_t)packet[34] << 8) |
packet[35]);

    data[2] = (((uint32_t)packet[36] << 24) | ((uint32_t)packet[37] << 16) | ((uint32_t)packet[38] << 8) |
packet[39]);

    return 0;
}

uint8_t MPU6050_6Axis_MotionApps20::dmpGetAccel(int16_t *data, const uint8_t* packet) {
    // TODO: accommodate different arrangements of sent data (ONLY default supported now)

```

```

    if (packet == 0) packet = dmpPacketBuffer;
    data[0] = (packet[28] << 8) | packet[29];
    data[1] = (packet[32] << 8) | packet[33];
    data[2] = (packet[36] << 8) | packet[37];
    return 0;
}

uint8_t MPU6050_6Axis_MotionApps20::dmpGetAccel(VectorInt16 *v, const uint8_t* packet) {
    // TODO: accommodate different arrangements of sent data (ONLY default supported now)
    if (packet == 0) packet = dmpPacketBuffer;
    v->x = (packet[28] << 8) | packet[29];
    v->y = (packet[32] << 8) | packet[33];
    v->z = (packet[36] << 8) | packet[37];
    return 0;
}

uint8_t MPU6050_6Axis_MotionApps20::dmpGetQuaternion(int32_t *data, const uint8_t* packet) {
    // TODO: accommodate different arrangements of sent data (ONLY default supported now)
    if (packet == 0) packet = dmpPacketBuffer;
    data[0] = (((uint32_t)packet[0] << 24) | ((uint32_t)packet[1] << 16) | ((uint32_t)packet[2] << 8) | packet[3]);
    data[1] = (((uint32_t)packet[4] << 24) | ((uint32_t)packet[5] << 16) | ((uint32_t)packet[6] << 8) | packet[7]);
    data[2] = (((uint32_t)packet[8] << 24) | ((uint32_t)packet[9] << 16) | ((uint32_t)packet[10] << 8) | packet[11]);
    data[3] = (((uint32_t)packet[12] << 24) | ((uint32_t)packet[13] << 16) | ((uint32_t)packet[14] << 8) | packet[15]);
    return 0;
}

uint8_t MPU6050_6Axis_MotionApps20::dmpGetQuaternion(int16_t *data, const uint8_t* packet) {
    // TODO: accommodate different arrangements of sent data (ONLY default supported now)
    if (packet == 0) packet = dmpPacketBuffer;
    data[0] = ((packet[0] << 8) | packet[1]);
    data[1] = ((packet[4] << 8) | packet[5]);
    data[2] = ((packet[8] << 8) | packet[9]);
    data[3] = ((packet[12] << 8) | packet[13]);
    return 0;
}

```

```

uint8_t MPU6050_6Axis_MotionApps20::dmpGetQuaternion(Quaternion *q, const uint8_t* packet) {
    // TODO: accommodate different arrangements of sent data (ONLY default supported now)

    int16_t qI[4];

    uint8_t status = dmpGetQuaternion(qI, packet);

    if (status == 0) {
        q -> w = (float)qI[0] / 16384.0f;
        q -> x = (float)qI[1] / 16384.0f;
        q -> y = (float)qI[2] / 16384.0f;
        q -> z = (float)qI[3] / 16384.0f;

        return 0;
    }

    return status; // int16 return value, indicates error if this line is reached
}

// uint8_t MPU6050_6Axis_MotionApps20::dmpGet6AxisQuaternion(long *data, const uint8_t* packet);
// uint8_t MPU6050_6Axis_MotionApps20::dmpGetRelativeQuaternion(long *data, const uint8_t* packet);
uint8_t MPU6050_6Axis_MotionApps20::dmpGetGyro(int32_t *data, const uint8_t* packet) {
    // TODO: accommodate different arrangements of sent data (ONLY default supported now)

    if (packet == 0) packet = dmpPacketBuffer;

    data[0] = (((uint32_t)packet[16] << 24) | ((uint32_t)packet[17] << 16) | ((uint32_t)packet[18] << 8) |
packet[19]);

    data[1] = (((uint32_t)packet[20] << 24) | ((uint32_t)packet[21] << 16) | ((uint32_t)packet[22] << 8) |
packet[23]);

    data[2] = (((uint32_t)packet[24] << 24) | ((uint32_t)packet[25] << 16) | ((uint32_t)packet[26] << 8) |
packet[27]);

    return 0;
}

uint8_t MPU6050_6Axis_MotionApps20::dmpGetGyro(int16_t *data, const uint8_t* packet) {
    // TODO: accommodate different arrangements of sent data (ONLY default supported now)

    if (packet == 0) packet = dmpPacketBuffer;

    data[0] = (packet[16] << 8) | packet[17];
    data[1] = (packet[20] << 8) | packet[21];
    data[2] = (packet[24] << 8) | packet[25];

    return 0;
}

uint8_t MPU6050_6Axis_MotionApps20::dmpGetGyro(VectorInt16 *v, const uint8_t* packet) {

```



```

// TODO: accommodate different arrangements of sent data (ONLY default supported now)
if (packet == 0) packet = dmpPacketBuffer;

v -> x = (packet[16] << 8) | packet[17];
v -> y = (packet[20] << 8) | packet[21];
v -> z = (packet[24] << 8) | packet[25];

return 0;
}

// uint8_t MPU6050_6Axis_MotionApps20::dmpSetLinearAccelFilterCoefficient(float coef);
// uint8_t MPU6050_6Axis_MotionApps20::dmpGetLinearAccel(long *data, const uint8_t* packet);
uint8_t MPU6050_6Axis_MotionApps20::dmpGetLinearAccel(VectorInt16 *v, VectorInt16 *vRaw,
VectorFloat *gravity) {

    // get rid of the gravity component (+1g = +8192 in standard DMP FIFO packet, sensitivity is 2g)
    v -> x = vRaw -> x - gravity -> x*8192;
    v -> y = vRaw -> y - gravity -> y*8192;
    v -> z = vRaw -> z - gravity -> z*8192;

    return 0;
}

// uint8_t MPU6050_6Axis_MotionApps20::dmpGetLinearAccelInWorld(long *data, const uint8_t* packet);
uint8_t MPU6050_6Axis_MotionApps20::dmpGetLinearAccelInWorld(VectorInt16 *v, VectorInt16 *vReal,
Quaternion *q) {

    // rotate measured 3D acceleration vector into original state

    // frame of reference based on orientation quaternion

    memcpy(v, vReal, sizeof(VectorInt16));

    v -> rotate(q);

    return 0;
}

// uint8_t MPU6050_6Axis_MotionApps20::dmpGetGyroAndAccelSensor(long *data, const uint8_t* packet);
// uint8_t MPU6050_6Axis_MotionApps20::dmpGetGyroSensor(long *data, const uint8_t* packet);
// uint8_t MPU6050_6Axis_MotionApps20::dmpGetControlData(long *data, const uint8_t* packet);
// uint8_t MPU6050_6Axis_MotionApps20::dmpGetTemperature(long *data, const uint8_t* packet);
// uint8_t MPU6050_6Axis_MotionApps20::dmpGetGravity(long *data, const uint8_t* packet);
uint8_t MPU6050_6Axis_MotionApps20::dmpGetGravity(int16_t *data, const uint8_t* packet) {

    /* +1g corresponds to +8192, sensitivity is 2g. */

    int16_t qI[4];

```

```

uint8_t status = dmpGetQuaternion(qI, packet);
data[0] = ((int32_t)qI[1] * qI[3] - (int32_t)qI[0] * qI[2]) / 16384;
data[1] = ((int32_t)qI[0] * qI[1] + (int32_t)qI[2] * qI[3]) / 16384;
data[2] = ((int32_t)qI[0] * qI[0] - (int32_t)qI[1] * qI[1]
           - (int32_t)qI[2] * qI[2] + (int32_t)qI[3] * qI[3]) / ((int32_t)(2 * 16384L));
return status;
}

uint8_t MPU6050_6Axis_MotionApps20::dmpGetGravity(VectorFloat *v, Quaternion *q) {
    v->x = 2 * (q->x*q->z - q->w*q->y);
    v->y = 2 * (q->w*q->x + q->y*q->z);
    v->z = q->w*q->w - q->x*q->x - q->y*q->y + q->z*q->z;
    return 0;
}

// uint8_t MPU6050_6Axis_MotionApps20::dmpGetUnquantizedAccel(long *data, const uint8_t* packet);
// uint8_t MPU6050_6Axis_MotionApps20::dmpGetQuantizedAccel(long *data, const uint8_t* packet);
// uint8_t MPU6050_6Axis_MotionApps20::dmpGetExternalSensorData(long *data, int size, const uint8_t*
packet);
// uint8_t MPU6050_6Axis_MotionApps20::dmpGetEIS(long *data, const uint8_t* packet);

uint8_t MPU6050_6Axis_MotionApps20::dmpGetEuler(float *data, Quaternion *q) {
    data[0] = atan2(2*q->x*q->y - 2*q->w*q->z, 2*q->w*q->w + 2*q->x*q->x - 1); // psi
    data[1] = -asin(2*q->x*q->z + 2*q->w*q->y); // theta
    data[2] = atan2(2*q->y*q->z - 2*q->w*q->x, 2*q->w*q->w + 2*q->z*q->z - 1); // phi
    return 0;
}

#ifdef USE_OLD_DMPGETYAWPITCHROLL
uint8_t MPU6050_6Axis_MotionApps20::dmpGetYawPitchRoll(float *data, Quaternion *q, VectorFloat
*gravity) {
    // yaw: (about Z axis)
    data[0] = atan2(2*q->x*q->y - 2*q->w*q->z, 2*q->w*q->w + 2*q->x*q->x - 1);
    // pitch: (nose up/down, about Y axis)
    data[1] = atan(gravity->x / sqrt(gravity->y*gravity->y + gravity->z*gravity->z));

```

```

// roll: (tilt left/right, about X axis)
data[2] = atan(gravity -> y / sqrt(gravity -> x*gravity -> x + gravity -> z*gravity -> z));
return 0;
}
#else
uint8_t MPU6050_6Axis_MotionApps20::dmpGetYawPitchRoll(float *data, Quaternion *q, VectorFloat
*gravity) {
// yaw: (about Z axis)
data[0] = atan2(2*q -> x*q -> y - 2*q -> w*q -> z, 2*q -> w*q -> w + 2*q -> x*q -> x - 1);
// pitch: (nose up/down, about Y axis)
data[1] = atan2(gravity -> x , sqrt(gravity -> y*gravity -> y + gravity -> z*gravity -> z));
// roll: (tilt left/right, about X axis)
data[2] = atan2(gravity -> y , gravity -> z);
if (gravity -> z < 0) {
if(data[1] > 0) {
data[1] = PI - data[1];
} else {
data[1] = -PI - data[1];
}
}
return 0;
}
#endif

// uint8_t MPU6050_6Axis_MotionApps20::dmpGetAccelFloat(float *data, const uint8_t* packet);
// uint8_t MPU6050_6Axis_MotionApps20::dmpGetQuaternionFloat(float *data, const uint8_t* packet);

uint8_t MPU6050_6Axis_MotionApps20::dmpProcessFIFOPacket(const unsigned char *dmpData) {
(void)dmpData; // unused parameter
/*for (uint8_t k = 0; k < dmpPacketSize; k++) {
if (dmpData[k] < 0x10) Serial.print("0");
Serial.print(dmpData[k], HEX);
Serial.print(" ");
}
}

```

```

Serial.print("\n");*/
//Serial.println((uint16_t)dmpPacketBuffer);

return 0;
}

uint8_t MPU6050_6Axis_MotionApps20::dmpReadAndProcessFIFOPacket(uint8_t numPackets, uint8_t
*processed) {
    uint8_t status;
    uint8_t buf[dmpPacketSize];
    for (uint8_t i = 0; i < numPackets; i++) {
        // read packet from FIFO
        getFIFOBytes(buf, dmpPacketSize);

        // process packet
        if ((status = dmpProcessFIFOPacket(buf)) > 0) return status;

        // increment external process count variable, if supplied
        if (processed != 0) (*processed)++;
    }
    return 0;
}

// uint8_t MPU6050_6Axis_MotionApps20::dmpSetFIFOProcessedCallback(void (*func) (void));

// uint8_t MPU6050_6Axis_MotionApps20::dmpInitFIFOParam();
// uint8_t MPU6050_6Axis_MotionApps20::dmpCloseFIFO();
// uint8_t MPU6050_6Axis_MotionApps20::dmpSetGyroDataSource(uint_fast8_t source);
// uint8_t MPU6050_6Axis_MotionApps20::dmpDecodeQuantizedAccel();
// uint32_t MPU6050_6Axis_MotionApps20::dmpGetGyroSumOfSquare();
// uint32_t MPU6050_6Axis_MotionApps20::dmpGetAccelSumOfSquare();
// void MPU6050_6Axis_MotionApps20::dmpOverrideQuaternion(long *q);
uint16_t MPU6050_6Axis_MotionApps20::dmpGetFIFOPacketSize() {
    return dmpPacketSize;
}

```

```
uint8_t MPU6050_6Axis_MotionApps20::dmpGetCurrentFIFOPacket(uint8_t *data) { // overflow proof
    return(GetCurrentFIFOPacket(data, dmpPacketSize));
}
```

KATA PENGANTAR

Puji dan syukur penulis ucapkan kepada Tuhan Yang Maha Esa, karena atas izinnya penulis dapat menyelesaikan penyusunan Instruction Manual Book Alat Peraga Sistem Water Ballast Otomatis Berbasis Arduino R3. Penyusunan manual book ini dapat digunakan sebagai panduan dalam mengoperasikan alat peraga sistem fresh water generator untuk taruna/taruni Politeknik Ilmu Pelayaran Semarang.

Penulis menyusun buku panduan ini untuk memudahkan para pengguna dalam mengenali dan mengetahui cara menggunakan alat peraga sistem water ballast otomatis. Penulis berharap semoga instruction manual book ini dapat bermanfaat serta berguna bagi taruna dan sivitas akademika PIP Semarang.

Semarang, Agustus 2023

FARIZ FAUZIAN

DAFTAR ISI

Kata Pengantar	1
Daftar Isi	2
Prosedur Penggunaan Alat Peraga Water Ballast Otomatis	3
A. Nama Alat Peraga	3
B. Fungsi Plat Peraga	3
C. Prosedur Menggunakan Alat Peraga	3
D. Foto Alat Peraga	7
E. Perawatan Alat Peraga.....	8
F. Gambaran Umum Prinsip Kerja Alat Peraga.....	8
G. Kegagalan Sistem dan Cara Mengatasi	8
H. Wire Diagram elektrik Sistem Kontrol Ballast Otomatis	9

PROSEDUR PENGGUNAAN ALAT PERAGA SISTEM *WATER BALLAST* OTOMATIS BERBASIS ARDUINO R3

A. Nama Alat Peraga

Sistem *Water Ballast* Otomatis Berbasis Arduino R3

B. Fungsi Alat Peraga



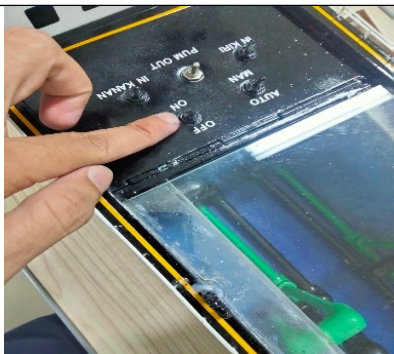

Sebagai alat pembelajaran bagi taruna agar memahami kegunaan *water ballast* di atas kapal yaitu, memberikan masa berat jenis air ke setiap tangki *ballast* yang akan diisi oleh air sebagai masa penyeimbang kapal yang bermuatan ataupun tidak bermuatan.

C. Prosedur Menggunakan Alat Peraga


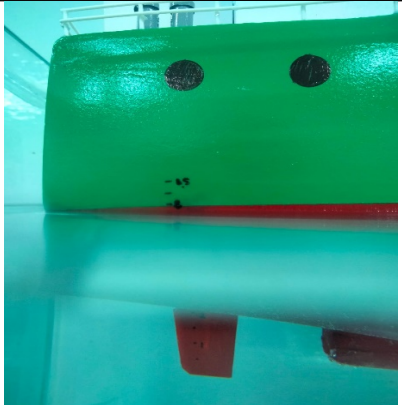


1. Pilih metode penggunaan *Power Adaptor* atau *Battery*.

Peringatan: jika ingin menggunakan *power adaptor* lepaskan *battery*, terlebih dahulu. agar tidak terjadi lonjakan arus listrik yang besar terhadap 2 power tersebut, yang nantinya akan merusak komponen kelistrikan lainnya.

NO	<u>PROSEDURE</u> <u>PENGOPERASIAN</u> <u>SECARA AUTO</u>	GAMBAR PROSEDUR
1.	Hidupkan power <i>switch</i> dari selektor <i>OFF</i> ke posisi <i>ON</i> pastikan panel instrumen atau LCD menyala.	
2.	Setelah <i>switch</i> dalam kondisi ON pastikan panel LCD menyala dan tunggu sekitar 5-10 dektik untuk memastikan gyro mengkalibrasi ulang Dan pastikan sudut kapal dalam kondisi seimbang	

3.	Pilih selektor AUTOMATIS atau MANUAL tergantung dengan kebutuhan pengguna	
4.	Dan alat siap digunakan berkerja secara AUTOMATIS	
NO	<u>PROSEDURE</u> <u>PENGOPERASIAN</u> <u>SECARA MANUAL</u>	GAMBAR PROSEDUR
1.	Hidupkan <i>power switch</i> dari selektor <i>OFF</i> ke posisi <i>ON</i> pastikan panel instrumen menyala.	
2.	Setelah <i>switch</i> dalam kondisi <i>ON</i> pastikan panel LCD menyala dan tunggu sekitar 5-10 dektik untuk memastikan gyro mengkalibrasi ulang Dan pastikan sudut kapal dalam kondisi seimbang.	

3.	Pilih selektor MANUAL tergantung dengan kebutuhan pengguna.	
4.	Pindah selektor <i>switch</i> pompa hisap <i>ballast</i> kanan dari posisi <i>OFF</i> ke <i>ON</i> , pastikan lampu indikator menyala sebagai pacuan indikasi bahwa pompa sedang bekerja. setelah merasa cukup pindahkan <i>switch</i> ke posisi <i>OFF</i> untuk menonaktifkan pompa.	
5.	Pindah selektor <i>switch</i> pompa hisap <i>ballast</i> kiri dari posisi <i>OFF</i> ke <i>ON</i> , pastikan lampu indikator menyala sebagai pacuan indikasi bahwa pompa sedang bekerja. setelah merasa cukup pindahkan <i>switch</i> ke posisi <i>OFF</i> untuk menonaktifkan pompa.	
6.	Pastikan kondisi ruangan sea chas kiri dan kanan tidak tersumbat, untuk memastikan proses hisap kepengisian <i>ballast</i> menjadi lancar.	
7.	Setelah tangki <i>ballast</i> sudah mencapai volume maksimal, operasikan pompa <i>ballast</i> over board untuk membuang air didalam tangki <i>ballast</i> untuk proses penyeimbangan ulang, setelah merasa cukup pindahan <i>switch</i> selektor ke <i>OFF</i> dan pastikan lampu indikator mati sebagai tanda bahwa pompa over board tidak bekerja.	

8.	<p>Proses <i>overboard ballast</i> untuk mengurangi volume air yang berada didalam tangka <i>ballast</i> sesuai dengan kebutuhan.</p>	
9.	<p>Ukur <i>draft</i> bagian belakang kapal (buritan) untuk memastikan muatan dan volume <i>draft</i> yang dibutuhkan untuk menjaga keseimbangan kapal.</p>	
10.	<p>Pengecekan <i>plimsollMark</i> untuk mengukur <i>draft</i> dibagian buritan, tengah dan haluan sebagai tolak ukur untuk mengetahui <i>draft</i> yang dibutuhkan.</p>	
11.	<p>Pengukuran <i>daftar</i> dibagian haluan dan memastikan kondisi kapal dalam keadaan stabil.</p>	

D. Foto Alat Peraga



In put Power
Adaptor



3 Motor Pump



Main Modul
Sensor Haluan



Selector Mode

Power Battery Area
Buritan

E. Perawatan Pada Alat Peraga

1. Bersihkan debu dengan lap.
2. Chek selalu kondisi motor pump dengan mengaktifkan seluruh komponen kelistrikan termasuk motor setidaknya seminggu sekali.
3. Buang sisa air yang berada di tangki *ballast*.
4. Hindari peletakan pada tempat lembab

F. Gambaran Umum Prinsip Kerja Alat Peraga

1. Hubungkan *Power Adaptor* atau *Battery* pada pengoprasian sistem *ballast* otomatis hidupakn Power ON pada Saklar dan sistem *ballast* siap dioperasikan secara AUTO atau MANUAL.
2. Pada sistem *ballast* otomatis terdapat sebuah panel LCD sebagai instrument sudut kemiringan kapal dan sebagai indikator bahwa sistem sedang beroperasi.

G. Kegagalan Sistem dan Cara Mengatasinya

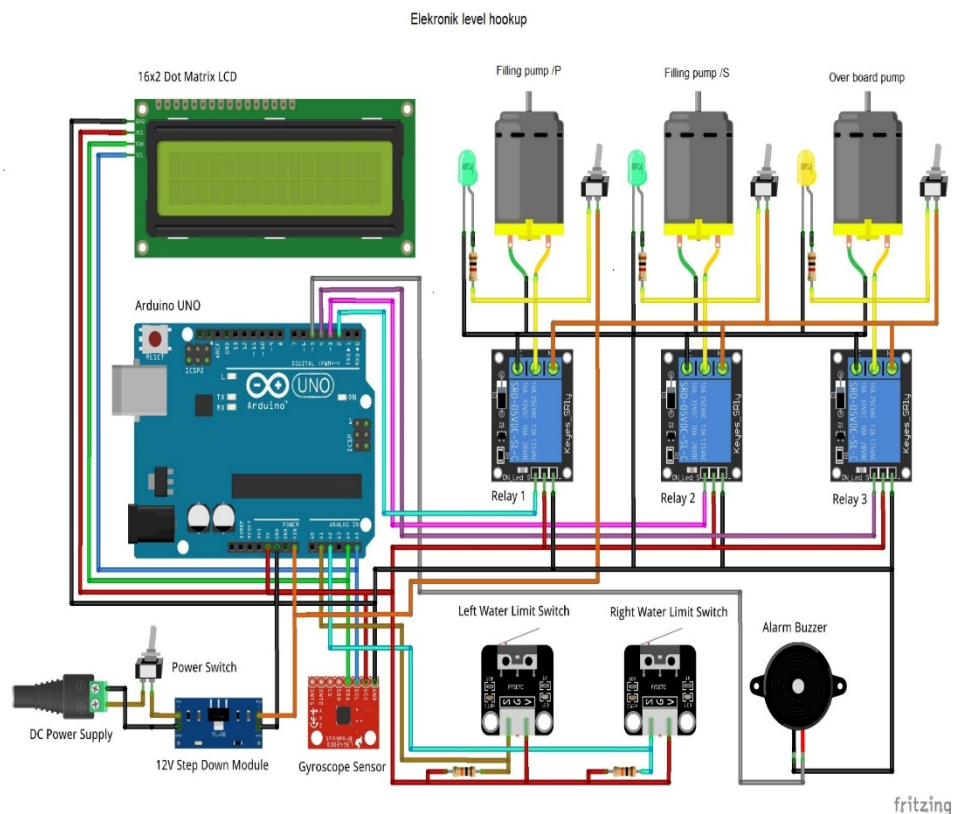
1. Panel instrument LCD tidak membaca kemiringan.

Cara mengatasi: pindahkan selector ke Power OFF dan kembalikan ke Power ON dan sistem akan berjalan kembali dengan semestinya.

2. Pompa Pengisian dan Pembuangan Tidak Dapat Beroperasi atau Berjalan Diluar Kendali

Cara mengatasi: pindahkan selector ke Power OFF kemudian pindahkan selector ke MODE MANUAL, aktifkan kembali Power ON, hidupkan Pompa Buang *ballast O/B*. pastikan air yang berada ditangi *ballast* terbuang, setelah dipastikan air yang berada didalam tangki *ballast* habis, kemudian matikan Pompa *ballast O/B*. Pindahkan kembali MODE selector dari MANUAL ke AUTO, sistem akan berjalan kembali dengan semestinya.

H. Wire Diagram Sistem Elektrik Alat Peraga *Water Ballast* otomatis



Keterangan:

1. D.C Power supply Adaptor/Battery
2. Rumah battery
3. Power Switch ON/OFF
4. 12V Step Down Module
5. ARDUINO R3
6. GYROSCOPE SENSOR 3050
7. 16x2 LCD ic2
8. Water Limit Sensor x2
9. Relly x3
10. Motor Pump x3
11. Alarm Buzzer
12. LED Indicator lamp x3