TUGAS AKHIR – ME184834

RANCANG BANGUN ALAT UKUR KETINGGIAN AIR LAUT DENGAN MENGGUNAKAN MODUL SENSOR GY 86

Dennis Chaniago Ramadhan
NRP 04211640000067

Dosen Pembimbing
Juniarko Prananda, S.T., M.T.
Ir. Agoes Ahmad Masroeri, M.Eng., D.Eng.

DEPARTEMEN TEKNIK SISTEM PERKAPALAN
FAKULTAS TEKNOLOGI KELAUTAN
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2020

"*Halaman Ini Sengaja Dikosongkan"*

TUGAS AKHIR – ME184834

RANCANG BANGUN ALAT UKUR KETINGGIAN AIR LAUT DENGAN MENGGUNAKAN MODUL SENSOR GY 86

Dennis Chaniago Ramadhan
NRP. 04211640000067

Dosen Pembimbing
Juniarko Prananda, S.T, M.T.
Ir. Agoes Ahmad Masroeri, M.Eng, D.Eng.

DEPARTEMEN TEKNIK SISTEM PERKAPALAN
FAKULTAS TEKNOLOGI KELAUTAN
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2020

"*Halaman Ini Sengaja Dikosongkan"*

FINAL PROJECT - ME184834

DESIGN OF SEA LEVEL MEASUREMENT WITH USING GY 86 MODULE

Dennis Chaniago Ramadhan
NRP. 04211640000067

Advisor :
Juniarko Prananda, S.T, M.T.
Ir. Agoes Ahmad Masroeri, M.Eng, D.Eng.

DEPARTMENT OF MARINE ENGINEERING
FACULTY OF MARINE TECHNOLOGY
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2020

"*Halaman Ini Sengaja Dikosongkan"*

LEMBAR PENGESAHAN

RANCANG BANGUN ALAT UKUR KETINGGIAN AIR LAUT DENGAN MENGGUNAKAN MODUL SENSOR GY 86

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

Oleh:

Dennis Chaniago Ramadhan
NRP 04211640000067

Disetujui oleh Pembimbing Tugas Akhir:

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

Ir. A.A Masroeri, M.Eng, D.Eng
NIP. 1958 0807 1984 03 1004

SURABAYA
Juli 2020

"*Halaman Ini Sengaja Dikosongkan"*

LEMBAR PENGESAHAN

RANCANG BANGUN ALAT UKUR KETINGGIAN AIR LAUT DENGAN MENGGUNAKAN MODUL SENSOR GY 86

TUGAS AKHIR

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

Penulis:

Dennis Chaniago Ramadhan
NRP. 04211640000067

Disetujui Oleh,
Kepala Departemen Teknik Sistem Perkapalan

Beny Cahyono, ST, MT, Ph.D.
NIP 197903192008011008

SURABAYA

AGUSTUS, 2020

"*Halaman Ini Sengaja Dikosongkan"*

RANCANG BANGUN ALAT UKUR KETINGGIAN AIR LAUT DENGAN MENGGUNAKAN MODUL SENSOR GY 86

Nama mahasiswa : Dennis Chaniago Ramadhan
NRP : 04211640000067
Pembimbing : 1. Juniarko Prananda, S.T., M.T
2. Ir. A. A. Masroeri, M.Eng, D.Eng

ABSTRAK

Perubahan ketinggian air laut merupakan pergerakan yang sangat dinamis. Perubahan tersebut dipengaruhi adanya gelombang air laut yang disebabkan oleh pergerakan angin dan juga aktivitas pasang surut air laut yang disebabkan oleh gaya gravitasi bulan dan matahari. Pergerakan air laut sangatlah erat dengan masyarakat dalam berbagai bidang seperti bidang geologi, biologi, transportasi serta energi. Untuk mendapatkan manfaat dari pergerakan air laut diperlukan data untuk analisa lebih lanjut. Data diperoleh dari sebuah alat ukur yang diletakkan langsung di permukaan (in-situ). Tetapi alat ukur yang ada di pasaran sangatlah mahal dan tidak fleksibel. Oleh karena itu tujuan penelitian ini adalah untuk mendesain alat ukur ketinggian air laut yang memiliki nilai ekonomis serta fleksibel dalam penggunaan.

Metode yang digunakan adalah dengan menggunakan perubahan tekanan atmosfir untuk mengukur ketinggian. Selain menggunakan sensor barometer, alat ukur ini juga dilengkapi dengan sensor temperatur udara dan air. Desain alat ukur ini dibagi menjadi beberapa system yaitu system penginderaan, sistem komunikasi, serta sistem catu daya. Beberapa komponen dalam sistem penginderaan adalah GY86, GPS Neo 6MV2, *Water Sensor* DS18B20, dan *Power Sensor* INA219. Untuk komponen komunikasi menggunakan dua buah modul radio NRF24L01 dan aplikasi Visul Studio untuk desain aplikasi server. Dan untuk sistem catu daya menggunakan *Charging Controller* TP4056 dan *DC-DC Step Up* XL6009.

Pengujian untuk ketinggian gelombang menggunakan simulator dengan simpangan maksimal sebesar 0.55 m. Sehingga dari data pengujian dapat diketahui error alat ukur sebesar 3.6% untuk pengukuran ketinggian, 11.9% untuk error pengukuran temperature udara dengan membandingkan hasil pengukuran dengan nilai temperatur udara yang dilihat dari *smartphone*, dan 1.6% untuk error dari hasil pengukuran tegangan baterai dengan membandingkan dengan nilai tegangan hasil pengukuran dari avometer digital.

Kata kunci: (Ketinggian Air Laut, GY 86, MS5611)

"*Halaman Ini Sengaja Dikosongkan"*

DESIGN OF SEA LEVEL MEASUREMENT USING GY 86 MODULE

Student Name	: Dennis Chaniago Ramadhan
NRP	: 04211640000067
Supervisor	: 1. Juniarko Prananda, S.T., M.T
	2. Ir. A. A. Masroeri, M.Eng, D.Eng

ABSTRACT

Changes in sea level is a very dynamic movement. The change is influenced by the sea waves caused by the movement of the wind and tidal activity caused by the gravitational force of the moon and the sun. The movement of sea water is very close to the community in various fields such as geology, biology, transportation, and energy. To get the benefits of seawater movements data is needed for further analysis. Data obtained from a measuring instrument placed directly on the sea surface (in-situ). But measuring devices on the market are very expensive and inflexible. Therefore, the purpose of this study is to design sea level height measuring devices that have economic value and are flexible in use.

The method used is to use changes in atmospheric pressure to measure altitude. In addition to using a barometer sensor, this gauge is also equipped with an air and water temperature sensor. The design of this measuring instrument is divided into several systems namely sensing systems, communication systems, and power supply systems. Some components in the sensing system are GY86, GPS Neo 6MV2, DS18B20 Water Sensor, and INA219 Power Sensor. The communication component uses two NRF24L01 radio modules and Visual Studio application for server application design. And for the power supply system using the TP4056 Charging Controller and DC-DC Step Up XL6009.

Test for wave height using a simulator with a maximum deviation of 0.55 m. So that the test data can be known error measuring instrument of 3.6% for height measurements, 11.9% for measurement errors in air temperature by comparing the measurement results with air temperature values seen from a smartphone, and 1.6% for errors from measuring battery voltage by comparing with values voltage measurements from a digital Avometer.

Keywords: ***Sea level, GY 86, MS5611***

"*Halaman Ini Sengaja Dikosongkan"*

KATA PENGANTAR

Assalamualaikum warahmatullahi wabarakatuh, Alhamdulillah dengan segala puji kehadirat Allah SWT yang telah memberikan rahmat serta hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir ini dengan judul **"RANCANG BANGUN ALAT UKUR KETINGGIAN AIR LAUT DENGAN MENGGUNAKAN MODUL SENSOR GY 86"** sebagai salah satu syarat dalam menyelesaiakan pendidikan sarjana di Departemen Teknik Sistem Perkapalan, Fakultas Teknologi Kelautan, Institut Teknologi, Sepuluh Nopember.

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

1. Ibu Cassa Siti Arum dan Bapak Sasongko selaku orang tua penulis yang selalu memberikan dukungan doa, motivasi, serta materi. Annisa Sasca Putri dan Galang Ramadhan Prakoso selaku adik-adik penulis yang selalu mendukung, menemani, dan mendoakan penulis.
2. Bapak Juniarko Prananda, S.T, M.T. dan Bapak Ir. A. A. Masroeri, M.Eng, D.Eng selaku dosen pembimbing Tugas Akhir penulis atas bimbingan ilmu dan arahannya kepada penulis.
3. Bapak Beny Cahyono, S.T., M.T., Ph.D. selaku kepala Departemen Teknik Sistem Perkapalan, FTK, ITS.
4. Bapak A. A. B. Dinariyana Dwi P., S.T., MES., Ph.D. selaku dosen wali yang selalu memberikan arahan dan motivasi selama penulis menempuh pendidikan di departemen ini.
5. Ustad dan Ustadzah Pondok Pesantren Darussalam Keputih Surabaya selaku guru yang selalu memberikan dukungan spiritual kepada penulis.
6. Keluarga besar Voyage 16 dan Laboratorium *Marine Electrical and Automation System* (MEAS) selaku keluarga terdekat dan tempat penulis mengembangkan diri selama berkuliah di ITS.
7. Dan semua pihak yang telah membantu menyelesaikan tugas akhir ini, yang tidak dapat penulis sebutkan satu persatu.

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

Surabaya, 20 Agustus 2020

Dennis Chaniago Ramadhan

"*Halaman Ini Sengaja Dikosongkan"*

DAFTAR ISI

LEMBAR PENGESAHAN v
LEMBAR PENGESAHAN vii
ABSTRAK ix
ABSTRACT xi
KATA PENGANTAR xiii
DAFTAR ISI xv
DAFTAR GAMBAR xix
DAFTAR TABEL xxi
BAB 1 PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah 2
1.3 Tujuan 2
1.4 Batasan Masalah 2
1.5 Konstribusi 2
BAB 2 DASAR TEORI 3
2.1 Kajian Penelitian Terkait 3
2.2 Pasang Surut 4
2.2.1 Interaksi Pasang Surut Bulan dan Matahari 4
2.2.2 Tipe Pasang Surut 5
2.2.3 Komponen Harmonik Pasang Surut 5
2.3 Mengukur Tekanan Absolut Menggunakan Elemen Pengindera Piezoresitif 6
2.3.1 Sensor Piezoresitif 6
2.3.2 Menentukan tekanan absolut menggunakan sensor piezoresitif 7
2.3.3 Perakitan sensor 8
2.3.4 Sinyal keluaran 9
2.3.5 Sinyal skala penuh 10
2.3.6 Aplikasi 10
2.4 Mengukur Ketinggian Dengan Menggunakan Sensor Tekanan MS5611 .. 11

2.4.1 Tekanan atmosfir 11
2.4.2 Prinsip altimeter 12
2.4.3 Pengukuran praktis 13
2.4.4 Sensor tekanan untuk altimeter 14
2.4.5 Sensor tekanan absolut MS5611 14
2.4.6 Sirkuit evaluasi sinyal 15
2.4.7 Korelasi pada chip sensor 15
2.5 Energi Gelombang Air Laut 16
2.5.1 Konsep energi gelombang air laut 16
2.5.2 Rumus energi gelombang air laut 16
2.6 Komponen Alat Ukur 17
2.6.1 Arduino Nano 17
2.6.2 Komponen Sensor 20
2.6.3 Komponen Pengolah Data 22
2.6.4 Komponen Pengirim Data 22
2.6.5 Komponen Sistem Catu Daya 23
BAB 3 METODOLOGI PENELITIAN 25
3.1 Langkah-Langkah Penelitian 25
3.1.1 Identifikasi dan Perumusan Masalah 25
3.1.2 Studi Literatur 26
3.1.3 Desain Sistem 26
3.1.4 Perencanaan Wiring Sistem 26
3.1.5 Perencanaan Bahasa Pemrograman 26
3.1.6 Perakitan Hardware 26
3.1.7 Integrasi Hardware dan Software 26
3.1.8 Pengujian Alat Ukur 26
3.1.9 Analisa Data 27
3.1.10 Hasil dan Kesimpulan 27
3.2 Timeline Penelitian 28
BAB 4 RANCANG BANGUN MODEL DAN PEMBAHASAN 29
4.1 Gambaran Umum 29

4.2 Desain Sistem Alat Ukur Ketinggian Air Laut 29
4.2.1 Sistem penginderaan 29
4.2.2 Sistem Pengiriman Data 30
4.3 Perencanaan Wiring Sistem 32
4.4 Perencanaan Bahasa Pemrograman 34
4.4.1 Program Induk 34
4.4.2 Program Untuk Aktivasi Sensor Gy 86 37
4.4.3 Program Untuk Aktivasi GPS 40
4.4.4 Program Pengiriman Data Melalui Radio 41
4.4.5 Program Manajemen Daya Baterai 44
4.4.6 Program untuk Sensor Temperatur Air 45
4.5 Perakitan Hardware 45
4.6 Integrasi Software dan Hardware 48
4.6.1 Membuat Program Tampilan *Dashboard* 48
4.6.2 Membuat Program Penyimpanan Database 59
4.6.3 Mengatur dan menyimpan database 61
4.7 Pengujian Alat Ukur 63
4.8 Analisa Hasil Pengujian 64
4.8.1 Analisa Hasil Pengujian Data Gelombang 64
4.8.2 Analisa Hasil Pengujian Temperatur Udara 65
4.8.3 Analisa Hasil Pengujian Voltase Baterai 66
4.9 Biaya Pembuatan 67
BAB 5 PENUTUP 68
5.1 Kesimpulan 68
A. Desain alat ukur 68
B. Biaya pembuatan alat ukur 68
5.2 Saran 68
DAFTAR PUSTAKA 70
PROFIL PENULIS 72
UCAPAN TERIMAKASIH 74

DAFTAR GAMBAR

Gambar 2.1 Perbandingan hasil pasang surut antara *wave buoy* dengan prediksi BIG3
Gambar 2.2 Atas : Pasang Surut Purnama, Bawah: Pasang Surut Perbani 4
Gambar 2.3 Tipe Pasut 5
Gambar 2.4 Diagram skematik dari sensor tekanan absolut 6
Gambar 2.5 Rangkaian elemen penginderaan silicon tipikal untuk menentukan tekanan absolut 7
Gambar 2.6 Elemen sensor tekanan silikon dalam bentuk wafer untuk pengukuran tekanan absolut (tampilan atas) pada tekanan atmosfer eksternal P 1. (P 1 >> P2) ...8
Gambar 2.7 Sensor tekanan absolut, menggunakan AMS 5812 sebagai contoh.....8
Gambar 2.8 Sirkuit jembatan Wheatstone.....9
Gambar 2.9 Diagram skematik dari sebuah sensor dengan sirkuit evaluasi 9
Gambar 2.10 Puncak Gunung Everest.....11
Gambar 2.11 Perubahan tekanan terhadap ketinggian 13
Gambar 2.12 Sirkuit prinsip MS5611 15
Gambar 2.2.13 Arduino Nano 17
Gambar 2.14 Layout Konfigurasi Pin Arduino Nano 19
Gambar 2.15 GY 86 20
Gambar 2.16 GPS NEO 6M 21
Gambar 2.17 Sensor Arus INA219 21
Gambar 2.18 *Multipexer* I^2C Adafruit TCA9548A 22
Gambar 2.19 Modul Radio NRF24L01.....22
Gambar 2.20 *Charge Controller* TP4056.....23
Gambar 2.21 *Boost Converter* XL6009.....23
Gambar 2.22 Baterai 18650 3.6V 3400 mAH.....24
Gambar 3.1 Diagram Alir Penelitian.....25
Gambar 3.2 Simulator *Wave Buoy* 27
Gambar 4.1 Blok Diagram Desain Alat Ukur Ketinggian Air Laut.....31
Gambar 4.2 Wiring skematik sistem alat ukur ketinggian air laut meggunakan software Eagle.....32
Gambar 4.3 Atas: layout pcb bagian bawah; Bawah: layout pcb bagian atas 33
Gambar 4.4 PCB tampak bawah 45
Gambar 4.5 PCB tampak atas.....46
Gambar 4.6 PCB dengan pin header 46
Gambar 4.7 Perangkat Pengirim *(Smart Buoy Client)* 47
Gambar 4.8 Perangkat Penerima *(Smart Buoy Server)* 47
Gambar 4.9 Tampilan awal dashboard.....57
Gambar 4.10 Tampilan dashboard setelah running.....58
Gambar 4.11 Tampilan dashboard ketika *client* dalam keadaan diam dan proses *charging.* 58
Gambar 4.12 Hasil pengujian alat ukur dalam bentuk database Mc.Acces 63
Gambar 4.13 Grafik Tinggi Gelombang (Wave Buoy vs Simulator).....64

Gambar 4.14 Pengukuran temperatur udara melalui *smartphone.* 65
Gambar 4.15 Pengukuran tegangan baterai menggunakan avometer digital..... 66

DAFTAR TABEL

Tabel 2.1 Tipe pasang surut berdasarkan bilangan Formzhal 6
Tabel 2.2 Konfigurasi Pin Arduino 18
Tabel 3.1 Timline pengerjaan tugas akhir 28
Tabel 4.1 Konfigurasi NRF24L01 untuk Arduino Nano.....30
Tabel 4.2 Data Pengujian Ketinggian Gelombang 64
Tabel 4.3 Hasil pegukuran temperatur udara selama pengujian.....65
Tabel 4.4 Hasil pengukuran tegangan baterai 66
Tabel 4.5 Rincian Biaya 67

"*Halaman Ini Sengaja Dikosongkan"*

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Perubahan ketinggian air laut merupakan pergerakan yang sangat dinamis. Pergerakan ini disebabkan oleh angin. Angin yang berada diatas permukaan air laut bergerak dan mentransfer energinya ke permukaan air yang menyebabkan riak-riak atau alunan yang biasa disebut sebagai gelombang air laut. Pergerakan gelombang inilah yang menyebabkan permukaan air laut selalu tidak rata. Meskipun dalam keadaan cuaca yang tenang hal ini cukup untuk menimbulkan riak-riak gelombang. Dan sebaliknya jika dalam keadaan dimana sedang terjadi badai besar maka bisa menimbulkan gelombang yang sangat besar dan bersifat merusak.

Selain pergerakan gelombang yang terus menerus, perubahan permukaan air laut juga disebabkan oleh adanya pasang surut. Pasang surut merupakan sebuah pergerakan naik atau turunnya permukaan air laut yang disebabkan oleh gravitasi bulan dan matahari. Aktivitas pasang surut sangatlah dekat dengan masyarakat dalam berbagai bidang seperti geologi, transportasi, biologi, serta pembangkit energi.

Dalam memperlajari sebuah pergerakan air laut dibutuhkan sebuah alat ukur yang kita kenal sebagai *wave buoy*. Sebuah alat ukur gelombang digunakan untuk menampilkan parameter gelombang berupa ketinggian, periode, cepat rambat gelombang, temperature air laut, kecepatan angin, serta daya yang dihasilkan dari pergerakan gelombang air laut.

Secara umum peralatan alat ukur gelombang memiliki dimensi yang besar serta biaya yang tidak murah. Dilansir dari sebuah website penjualan *wave buoy*, sebuah alat ukur gelombang dijual dengan harga €60,900 atau sekitar 1 milliar rupiah [1]. Hal ini tentunya akan menjadi sebuah hambatan bagi para peneliti untuk mendapatkan data gelombang. Sehingga salah satu hal yang melatarbelakangi disusunnya karya tulis ini adalah untuk menciptakan sebuah alat ukur gelombang air laut yang fleksibel dengan harga terjangkau.

1.2 Rumusan Masalah

Berikut merupakan rumusan masalah yang akan dibahas dalam penelitian ini:

1.2.1 Bagaimana desain alat ukur ketinggian air laut dengan modul sensor GY 86?
1.2.2 Berapa total biaya pembuatan sistem elektronik alat ukur ketinggian air laut dengan menggunakan modul sensor GY 86?

1.3 Tujuan

Berikut merupakan tujuan dilakukannya penelitian ini:

1.3.1 Mendesain alat ukur gelombang air laut dengan menggunakan modul sensor GY 86.
1.3.2 Mengetahui berapa banyak biaya yang digunakan untuk membangun sistem elektronik alat ukur ketinggian air laut dengan menggunakan modul sensor GY 86.

1.4 Batasan Masalah

Adapun batasan masalah pada penelitian ini adalah sebagai berikut:

1.4.1 Tidak memperhitungkan desain dan stabilitas dari bagan apung alat ukur.
1.4.2 Pengujian alat dilakukan dengan menggunakan simulator sederhana.
1.4.3 Data pada penelitian ini hanya ditransfer melalui modul radio sehingga hanya bisa ditampilkan pada sebuah perangkat komputer yang sudah disambungkan dengan modul *receiver.*

1.5 Konstribusi

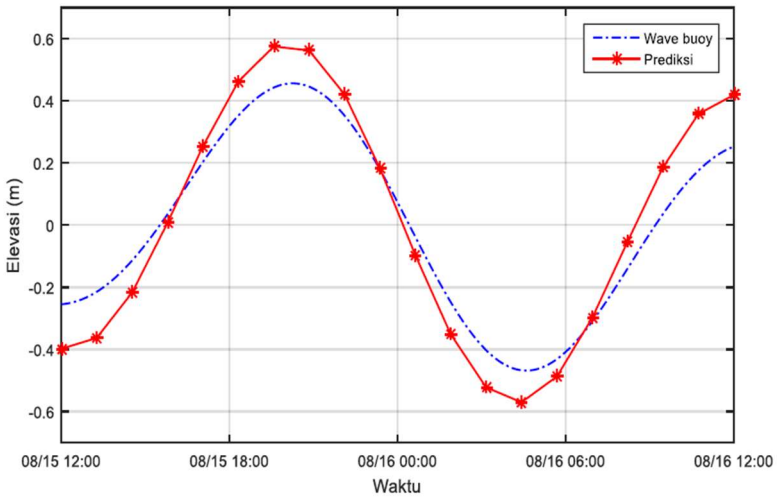
Berikut merupakan manfaat atau kontribusi yang dapat diambil dari penelitian ini:

1.5.1 Bagi institusi penelitian ini diharapkan dapat menambah kekayaan teknologi pada bidang kelautan.
1.5.2 Bagi peneliti penelitian sebagai perluasan pengetahuan terutama pada bidang teknologi kelautan.
1.5.3 Bagi masyarakat luas penelitian ini dapat dikembangkan sebagai suatu teknologi tepat guna untuk mengetahui data perubahan ketinggian air laut dan bisa dikembangkan untuk data PLTG.

BAB 2
DASAR TEORI

2.1 Kajian Penelitian Terkait

Dalam sebuah penelitian yang berjudul Rancang Bangun Dan Uji Kinerja Wave Buoy Sebagai Alat Pengukur Tinggi Gelombang Pesisir**,** (Erik Munandar dkk, 2018) mendesain alat pengukur tinggi gelombang yang diberi nama *wave buoy* yang mampu mengukur berbagai macam jenis gelombang air laut dengan menggunakan sensor *accelerometer* ADXL 345, Grove RTC DS 1307 sebagai penanda waktu dan Arduino Mega 2560 sebagai mikro kontroler. Hasilnya *wave buoy* dapat mengukur gelombang dengan galat sebesar 0.01-0.07 m. [2]



Gambar 2.1 Perbandingan hasil pasang surut antara *wave buoy* dengan prediksi BIG

Sumber : Erik Munandar dkk. 2018

Pada penelitian lain juga telah dibuat sebuah sistem alat pemantau ketinggian permukaan air yang dapat digunakan untuk memantau fenomena pasang surut air laut di Teluk Kendari dengan menggunakan jaringan internet. Perangkat keras atau alat yang digunakan terdiri dari sensor ultrasonik yang berfungsi untuk mengukur jarak permukaan air, mikrokontroler Arduino Uno yang berfungsi untuk mengolah data hasil pembacaan dari sensor ultrasonik, dan modul WiFi ESP8266 ESP-01 yang berfungsi menghubungkan mikrokontroler ke jaringan internet (data server). Perangkat keras dari sistem ini diujicobakan pada permukaan air yang berada di dalam sebuah wadah penampukan air, dimana permukaan air yang berada di dalam wadah tersebut diasumsikan sebagai permukaan air laut. Pengujian dilakukan untuk mengetahui kelayakan alat tersebut sebelum diterapkan di kawasan Teluk Kendari. Data hasil pengukuran ketinggian permukaan air dapat diakses oleh siapapun melalui browser baik menggunakan komputer (PC), laptop, maupun smartphone yang

terhubung dengan jaringan internet. Aplikasi yang digunakan untuk memantau ketinggian permukaan air melalui jaringan internet adalah ThingSpeak yang terdapat di website www.thingspeak.com.

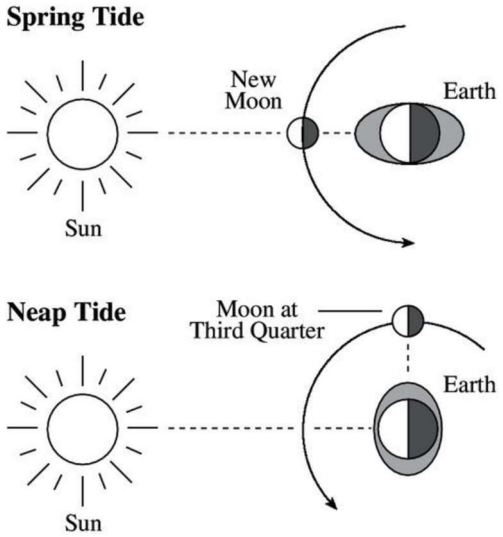
2.2 Pasang Surut

Pasang surut adalah fluktuasi muka air aut karena adanya gaya tarik benda benda di langit, terutama matahari dan bulan terhadap masa air laut di bumi. Meskipun masa bulan jauh lebih kecil dari masa matahari, tetapi karena jaraknya terhadap bumi lebih dekat, maka pengaruh gaya tarik bulan terhadap bumi lebih besar daripada pengaruh gaya tarik matahari. Gaya tarik bulan yang mempengaruhi pasang surut adalah 2,2 kali lebih besar daripada gaya tarik matahari. [3].

Pengetahuan tentang pasang surut adalah penting didalam perencanaan bangunan pantai dan pelabuhan. Elevasi muka air tertinggi (pasang) dan terendah (surut) sangat penting untuk merencanakan bangunan bangunan tersebut. Sebagai contoh, elevasi puncak bangunan pemecah gelombang dan dermaga ditentukan oleh elevasi muka air pasang, sementara kedalaman alur pelayaran/ pelabuhan ditentukan oleh muka air surut [3].

2.2.1 Interaksi Pasang Surut Bulan dan Matahari

Interaksi pasang surut bulan dan matahari dibagi menjadi dua, yaitu pasang surut purnama dan pasang surut perbani. Pasang surut purnama merupakan pasang surut dimana posisi bumi bulan dan matahari sejajar. Keadaan ini menyebabkan terjadinya pasang naik lebih tinggi dan surut lebih rendah. Pasang surut ini terjadi pada saat bulan baru dan bulan purnama. Pasang perbani merupakan pasang surut yang terjadi pada saat bulan, bumi dan matahari membentuk sudut 90^o dan 270^o . Bulan dikatakan dalam keadaan seperempat bagian ketika pasang surut perbani [4]. Pasang purnama dan pasang perbani yang dibentuk oleh posisi bulan dan matahari terhadap bumi seperti terlihat pada **Gambar 2.2**



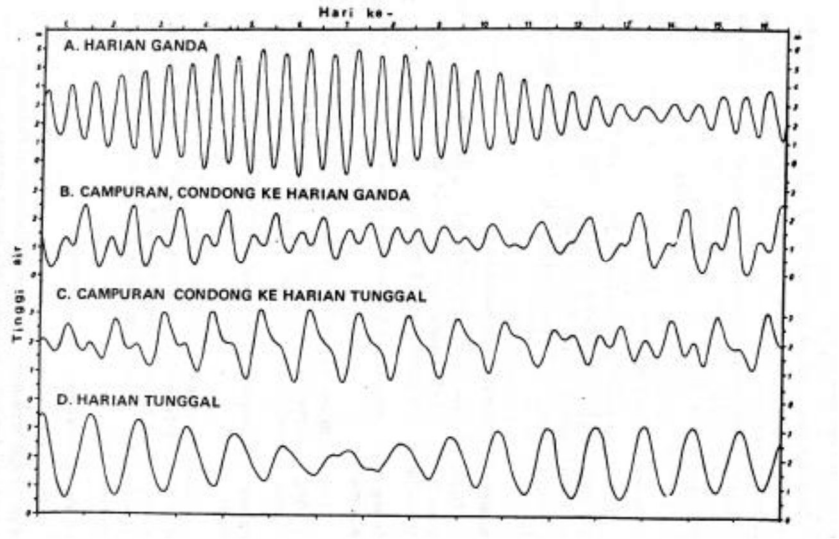
Gambar 2.2 Atas : Pasang Surut Purnama, Bawah: Pasang Surut Perbani

Sumber: Hicks 2006

2.2.2 Tipe Pasang Surut

Pada umumnya pasang surut memiliki empat tipe, yaitu [5]:

1. Pasang surut harian tunggal(Diurnal Tide) merupakan pasang surut yang hanya terjadi satu kali pasang dan satu kali surut dalam satu hari sebagai contoh terdapat di Selat Karimata.
2. Pasang surut harian ganda (Semi Diurnal Tide) merupakan pasang surut yang terjadi dua kali pasang dan dua kali surut yang tingginya hampir sama dalam satu hari sebagai contoh terdapat di Selat Malaka sampai Laut Andman.
3. Pasang surut campuran condong harian tunggal (Mixed Tide, Prevailing Diurnal) merupakan pasang surut yang tiap harinya terjadi satu kali pasang dan satu kali surut tetapi terkadang dengan dua kali pasang dan dua kali surut yang sangat berbeda dalam tinggi dan waktu terdapat di Pantai Selatan Kalimantan dan Pantai Utara Jawa Barat.
4. Pasang surut campuran condong harian ganda (Mixed Tde, Prevailing Semi Diurnal) merupakan pasang surut yang terjadi dua kali pasang dan dua kali surut dalam sehari tetapi terkadang terjadi satu kali pasang terdapat di Pantai Selatan Jawa dan Bagian Timur.



Gambar 2.3 Tipe Pasut

Sumber: Hicks 2006

2.2.3 Komponen Harmonik Pasang Surut

Rotasi bumi, revolusi bumi terhadap matahari dan revolusi bulan terhadap bumi menyebabkan resultan gaya penggerak pasang surut yang rumit dapat diuraikan sebagai hasil gabungan sejumlah komponen harmonik pasang surut (*Harmonic Constituent*). Komponen harmonik tersebut dapat dibagi menjadi tiga komponen yaitu tengah harian, harian dan periode panjang [6]. Beberapa komponen harmonik yang penting dapat dilihat, tipe pasang surut di suatu perairan dapat diketahui dengan

menggunakan persamaan yang biasa disebut dengan bilangan Formzhal [7] dan klasifikasi dari tipe pasang surut berdasarkan bilangan Formzhal dapat dilihat pada **Tabel 2.1**

Tabel 2.1 Tipe pasang surut berdasarkan bilangan Formzhal

Bilangan Formnzhal	Tipe Pasang Surut
0-0.25	*semidunal*
0.25-1.5	*Mixed, mainly semidunal*
1.5-3	*Mixed, mainly diurnal*
>3	*diurnal*

Sumber: (Pugh, 1987)

$$F = \frac{Hk1 + Ho1}{Hm2 + Hs2} \quad (1)$$

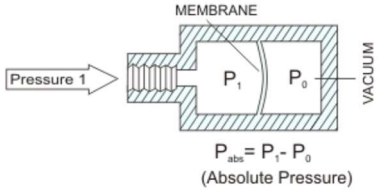
Keterangan:
Hk1 = Luni-Solar Diurnal
Ho1 = Prinsip Lunar Diurnal
Hm2 = Prinsip Lunar Semi - Diurnal
Hs2 = Prinsip Solar Semi - Diurnal

2.3 Mengukur Tekanan Absolut Menggunakan Elemen Pengindera Piezoresitif

2.3.1 Sensor Piezoresitif

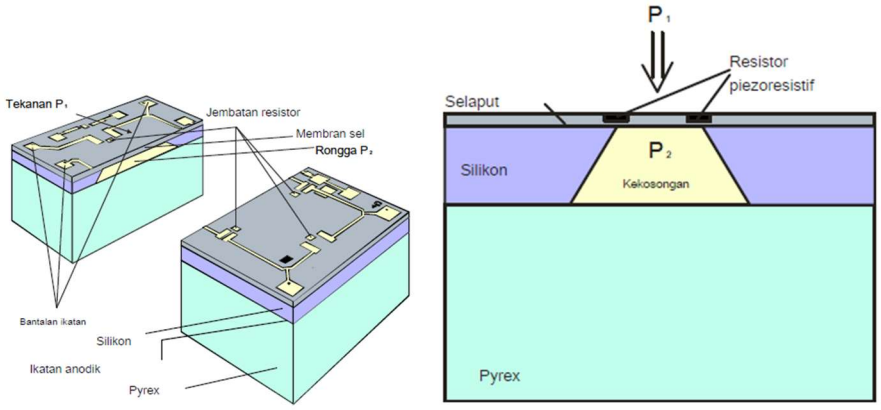
Dalam teknologi sensor, beberapa metode berbeda digunakan untuk mengukur tekanan. Biasanya dibedakan antara pengukuran tekanan relative, differensial, dan absolut. Pada modul sensor GY86 terdapat sensor MS5611 yang merupakan sensor piezoeresistif untuk menentukan tekanan absolut.

Mengukur tekanan absolut adalah pengukuran tekanan P1 relatif terhadap vakum sempurna P0 = 0 bar.



Gambar 2.4 Diagram skematik dari sensor tekanan absolut

Karena elemen konverter mikromekanis berbasis silikon diproduksi menggunakan teknologi semikonduktor (lihat Gambar 2.5), mereka memenuhi tuntutan tinggi yang dibuat dari keandalan dan ekonomi yang merupakan keunggulan dari teknologi ini. Semua elemen penginderaan tekanan mikromekanis yang terbuat dari silikon memiliki membran tipis sebagai elemen peka tekanan yang terukir secara anisotropis dari chip silikon (blok silikon), membentuk rongga. Pada titik yang sesuai pada membran, atom asing lokal ditanamkan dalam kristal silikon menggunakan teknologi semikonduktor, menciptakan zona dengan konduktivitas listrik yang berubah yang memiliki sifat resistor. Segera setelah tekanan diterapkan pada membran, struktur molekul kristal mengalami deformasi ketika membran silikon tipis membelok. Khususnya pada posisi resistor ada deformasi yang ditandai pada kristal yang mengarah pada perubahan yang terukur pada nilai resistifnya (efek piezoresistif).



Gambar 2.5 Rangkaian elemen penginderaan silicon tipikal untuk menentukan tekanan absolut

2.3.2 Menentukan tekanan absolut menggunakan sensor piezoresitif

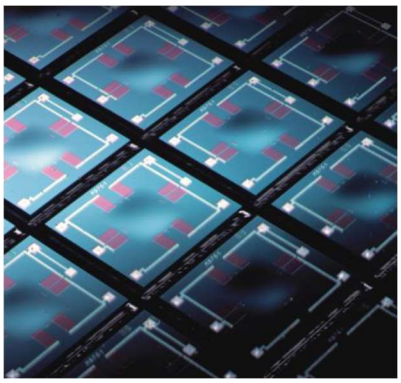
Saat mengukur tekanan absolut (lihat Gambar 2.5) tekanan pengukuran P1 direkam sehubungan dengan tekanan referensi P2 di dalam rongga, yang harus sangat rendah agar dapat diabaikan dibandingkan dengan tekanan yang akan diukur. Dalam pengaturan yang ideal, P2 akan sama dengan 0 bar (mis. P2 akan menjadi ruang hampa sempurna).

Contoh numerik. Pengukuran tekanan absolut 700 mbar menunjukkan tekanan pengukuran 700 mbar di atas vakum sempurna dan 313,25 mbar di bawah tekanan normal. (Tekanan normal = 1.013,25 mbar di permukaan laut, 0 ° C, dan garis lintang 45 °: dengan kata lain, tekanan negatif yang dapat dilihat dibandingkan dengan tekanan normal.)

Konsekuensi dari ini untuk sensor tekanan absolut (Gambar 2.5) adalah bahwa elemen penginderaan harus diproduksi dalam ruang hampa. Lebih tepatnya, agar tekanan negatif P2 sesuai elemen penginderaan harus ditutup rapat dengan substrat Pyrex. Ini dilakukan dalam proses elektrokimia yang dikenal sebagai ikatan anodik.

Untuk stabilitas pengukuran, tekanan negatif P2 (tekanan referensi) harus secara permanen mempertahankan nilai yang sama selama masa pakai sensor.

Saat tekanan P 1 diterapkan pada permukaan atas membran, ini membungkuk ke arah tekanan negatif. Sebagai P1 >> P2 berlaku, membran dengan demikian melengkung ke dalam ke dalam rongga vakum (lihat Gambar 2.6).

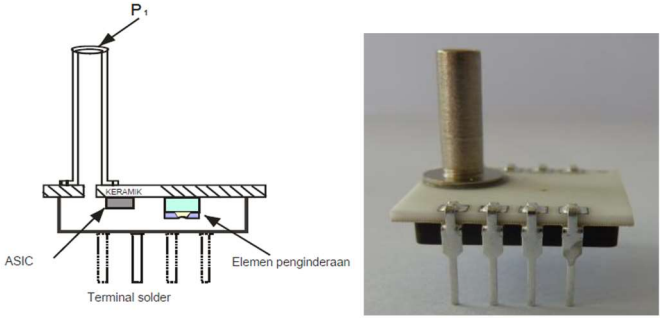


Gambar 2.6 Elemen sensor tekanan silikon dalam bentuk wafer untuk pengukuran tekanan absolut (tampilan atas) pada tekanan atmosfer eksternal P 1. (P 1 >> P2)

Pada Gambar 2.6: Struktur putih pada permukaan elemen sensor tekanan silikon adalah garis dan bantalan aluminium (kotak putih) yang digunakan untuk menghubungkan kabel penghubung ke sirkuit luar. Persegi panjang ungu tersebar, konektor konduktif untuk resistor piezoresistif. Resistor piezoresistif tersebar rendah (tidak terlihat di sini) dengan demikian terletak di antara area ungu di tepi defleksi di mana ada ketegangan mekanis terbesar. Di tengah elemen penginderaan deformasi membran terlihat, disebabkan oleh aplikasi eksternal tekanan atmosfer P1.

2.3.3 Perakitan sensor

Sensor tekanan absolut piezoresistif terdiri dari elemen penginderaan yang dipasang pada substrat, amplifier dan elektronik evaluasi (ASIC), dan rumahan dengan tabung dan koneksinya untuk pengkondisian sinyal listrik berikut.



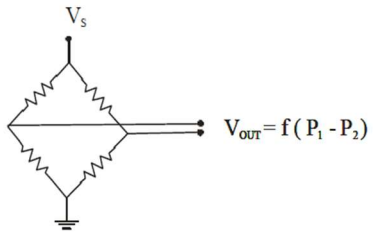
Gambar 2.7 Sensor tekanan absolut, menggunakan AMS 5812 sebagai contoh

2.3.4 Sinyal keluaran

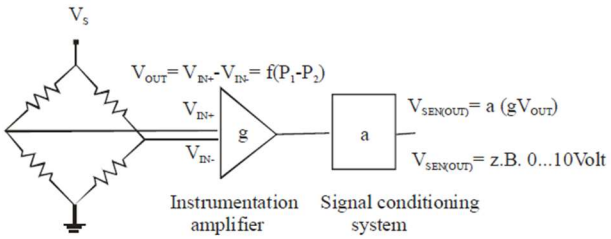
Efek piezoresistif elemen penginderaan monokristalin menyebabkan sinyal, yang sebanding dengan tekanan aktif pada keluaran jembatan Wheatstone (ditanam) Gambar 2.5). Hal berikut ini berlaku (S = sensitivitas):

$$Vout \approx S.(P1 - P2) \quad (2)$$

Sinyal diferensial ini diperkuat oleh faktor g dalam amplifier instrumentasi sistem pengkondisian sinyal (Gambar 2.9). Dengan proporsionalitas peningkatan karakteristik transfer sensor ditentukan; offset dan titik akhir belum diperbaiki, yaitu faktor konversi a harus ditetapkan dengan kalibrasi yang sesuai.



Gambar 2.8 Sirkuit jembatan Wheatstone



Gambar 2.9 Diagram skematik dari sebuah sensor dengan sirkuit evaluasi

Ketika P1 = P2, tegangan Vout pada output sensor jembatan biasanya digambarkan sebagai tegangan offset. Ini berlaku tanpa batasan untuk semua sensor tekanan. Ketika, menurut definisi, tekanan absolut harus diukur dalam kaitannya dengan ruang hampa sempurna, rongga harus memiliki nilai Po.

Namun, tekanan referensi 0 bar di rongga secara teknis tidak memungkinkan. Dalam praktiknya tekanan P2 hadir sehingga lebih besar dari 0 bar dalam semua kasus. Dengan mempertimbangkan definisi tekanan absolut, tekanan P2 ≠ 0 bilah menghasilkan offset yang mendistorsi pengukuran. Selama kalibrasi offset nilai ini dihitung menggunakan algoritma dan disesuaikan secara elektronik dalam sistem pengkondisian sinyal (Gambar 2.9). Untuk tujuan ini, kurva tekanan karakteristik dari sensor yang diukur secara individual dicatat pada dua tekanan, seperti tekanan nominal dan tekanan rendah, misalnya. Garis lurus melalui dua titik ini diekstrapolasi hingga mencapai nilai P2 = 0. Bagian sumbu y yang dihasilkan setara dengan

tegangan offset yang harus dikalibrasi dengan nilai offset yang diperlukan, misalnya 0 V.

2.3.5 Sinyal skala penuh

Sinyal skala penuh atau Full-sclae output (FSO) didefinisikan sebagai sinyal keluaran pada tekanan pengukuran maksimum dikurangi tegangan offset. Dengan sinyal keluaran sensor misalnya 10 V dan tegangan offset 0,05 V rentangnya 9.95 V. FSO tidak harus dengan sinyal skala penuh atau FS yang didefinisikan sebagai tekanan pengukuran termasuk offset dan yang 10 V dalam contoh yang diberikan. Seperti halnya offset, sinyal skala penuh harus dikalibrasi ke nilai output akhir yang diperlukan, seperti misalnya 10 V atau nilai lainnya.

Dalam sensor tekanan absolut rentang selalu mengacu pada ruang hampa sempurna. Menentukan kalibrasi tegangan offset dan sinyal bentang memberikan karakteristik transfer sensor pada suhu kamar. Karena sinyal offset dan bentang bergantung pada suhu, kesalahan ini harus dikompensasi. Untuk tujuan ini, selama kompensasi, kesalahan suhu offset (TSO) dan span (TCS) diukur pada berbagai suhu dan diperbaiki dalam sistem pengkondisian sinyal dengan cara yang sama seperti offset dan span. Dengan sensor atau pemancar yang diperkuat, kedua prosedur ini (kalibrasi dan kompensasi) dilakukan oleh pabrikan.

2.3.6 Aplikasi

Di pabrik industri, di mana tekanan dalam suatu sistem harus dipantau dan di mana tekanan sekitar tidak berpengaruh pada tekanan dalam sistem, seperti dalam sistem udara tertutup, misalnya, sensor tekanan absolut adalah cara yang sesuai untuk merekam tekanan. Salah satu contoh umum dari penggunaan sensor tekanan absolut adalah dalam pengukuran tekanan di tabung gas atau dalam sistem pneumatik (kompresor). Keuntungan dari sensor tekanan absolut dalam sistem tertutup tersebut terletak pada konstruksinya yang relatif sederhana. Berbeda dengan sensor tekanan relatif, tidak ada koneksi tambahan yang harus dibuat antara sensor dan lingkungannya; sensor tekanan absolut hanya harus dihubungkan ke bagian dalam sistem yang akan diukur. Contoh aplikasi lain adalah sistem yang terdiri dari pompa dan stop kontak yang terhubung (ruang vakum) yang akan dievakuasi dan dipantau. Tergantung pada seberapa kedap udara pompa dan sistem, tekanan negatif dapat dibuat di ruang vakum yang, bagaimanapun, hanya dicatat dengan benar hingga nilai P2 hadir di rongga.

Salah satu aplikasi populer pengukuran tekanan absolut adalah pencatatan tekanan barometrik misalnya antara 700 dan 1.200 mbar sebagai tekanan ambien efektif. Dalam contoh ini offset dikalibrasi ke 700 mbar dan sinyal skala penuh menjadi 1.200 mbar untuk memastikan dinamika tampilan optimal. Dengan pengukuran tekanan barometrik, dapat dengan mudah menentukan ketinggian dengan persamaan sebagai berikut [10]:

$$h = \frac{288{,}15\,\text{K}}{0{,}0065\,\text{K/m}}\left[1-\left(\frac{p}{101.325\,\text{Pa}}\right)^{0{,}0065\,\text{K/m}\frac{R}{g}}\right] \quad (3)$$

Pendekatan rumus diatas dengan memperhatikan asumsi sebagai berikut:

Tekanan pada permukaan laut = 101325 Pa = 1013.25 mbar

Temperatur pada permukaan laut = 288.15 K

Gradien temperature = 6.5 K/km

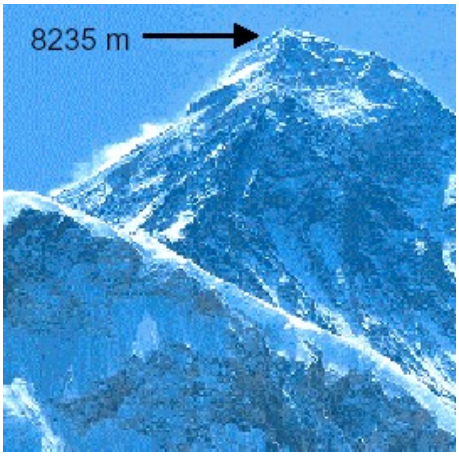
Konstanta gas spesifik (R) = R*/Mo (R = 287.052 J/K kg)

Percepatan gravitasi (g) = 9.80620 m/s^2 pada 45^o garis lintang

2.4 Mengukur Ketinggian Dengan Menggunakan Sensor Tekanan MS5611

2.4.1 Tekanan atmosfir

Tekanan atmosfir didefinisikan sebagai tekanan yang timbul akibat dari berat udara yang ada di sekitar lingkungan. Tekanan udara akan selalu berkurang akibat kenaikan ketinggian dari titik referensinya pada permukaan air laut (1013 mbar). Diukur dari permukaan air laut, tekanan atmosfir berubah sekitar 1 mbar/8 m. Ini hanyalah suatu asumsi pendekatan karena ada hubungan yang tidak linear antara tekanan atmosfir dan ketinggian karena kompresibiltas udara. Semisal di puncak Gunung Everest dengan ketinggian 8848 m dari permukaan air laut, maka tekanan atmosfirnya adalah 310 mbar.



Gambar 2.10 Puncak Gunung Everest

2.4.2 Prinsip altimeter

Suatu rumusan dasar barometric yang tidak memperhatikan hubungan perubahan temperature atau pada kondisi isothermal adalah sebagai berikut:

$$p(h_1) = p(h_0)e^{-\frac{Mg}{RT}\Delta h} \tag{4}$$

Dimana:

R = konstanta gas universal (8.314 J/K.mol

M = rata-rata molaritas massa gas pada atmosfir (0.02896 kg/mol)

g = percepatan gravitasi pada 45^o garis lintang (9.80620 m/s^2)

T = Temperatur absolut (273K)

Δh = h_1-h_0

Padahal suatu temperature mengalami perubahan akibat perubahan ketinggian, namun mengasumsikan bahwa hubungan linear antara temperature dengan ketinggian akan sangat terbatas pada trophosfer, maka persamaan berikut menyatakan perubahan temperature T(h):

$$T(h) = T(h_0) - a \,.\, (h_1-h_0) \tag{5}$$

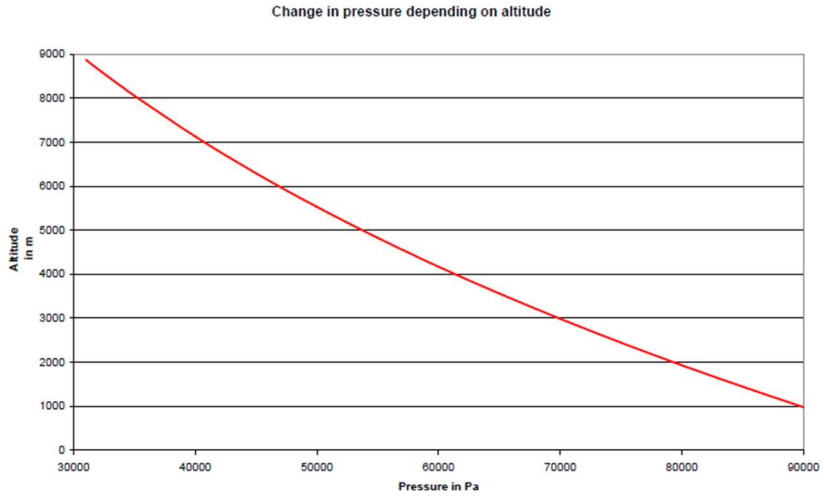
Dengan memasukan persamaan 6 ke persamaan 5 maka persamaan barometric menjadi:

$$p(h_1) = p(h_0)\left(1 - \frac{a\Delta h}{T(h_0)}\right)^{\frac{Mg}{Ra}} \tag{6}$$

Mengukur temperature pada atmosfir dengan hubungan linear menunjukkan hasil yang baik pada kondisi cuaca normal, dimana dengan mengasumsikan nilai a (gradien temperature) rata-rata adalah 0.65 K/100m. Dengan R yang konstan, M dan g yang diasumsikan konstan maka persamaan tersebut akan menghasilkan ketidak akuratan yang minor.

Secara teori, hubungan fungsi antara tekanan atmosfir dengan ketinggian (lihat Gambar 2.11) menunjukkan bahwa kita dapat menggunakan tekanan absolut sebagai altimeter atau alat ukur ketinggian. Untuk itu kita tetap harus memperhatikan bahwa sensor tetap harus dikalibrasi dengan skala ukur tertentu. Sehingga sinyal output dari

sensor akan proporsional dengan perubahan atmosfir dan sinyal keluaran maksimal (FSO) akan stabil.



Gambar 2.11 Perubahan tekanan terhadap ketinggian

2.4.3 Pengukuran praktis

Jika suatu sensor tekanan absolut digunakan untuk mengukur tekanan atmosfir pada titik tertentu, kita harus memperhatikan keterkaitan antara perubahan ketinggian dengan tekanan barometric. Artinya kita selalu mengukur perbedaan efek ketika mengukur tekanan atmosfir. Berikut merupakan beberapa metode pengukuran ketinggian berdasarkan kondisi yang ada.

2.4.3.1 Ketinggian h0 diketahui pada titik awal dari pengukuran

Pada ketinggian h_0 yang merupakan ketinggian geografi diketahui, kita dapat menentukan tekanan yang ada pada ketinggian tersebut $p(h_0)$ dan temperature $T(h_0)$. Dan tinggal memasukan ke persamaan 7 bersamaan dengan nilai h_0. Selanjutnya untuk tekanan $p(h_1)$ yang diukur pada ketinggian h_1 nilainya juga dimasukan ke dalam persmaan 7. Sehingga h_1 dapat diketahui nilainya melalui persamaan $\Delta h = h_1 - h_0$.

2.4.3.2 Ketinggian h0 tidak diketahui pada titik awal pengukuran

Pada kondisi seperti ini, nilai h_0 adalah yang pertama dicari. Sehingga kita memerlukan sebuah tekanan barometric pada kondisi normal (peta isobar) dan temperature yang relevan. Nilai tersebut lalu dimasukan ke dalam persamaan 7 dan tekanan $p(h_1)$ diukur pada titik awal pengukuran ketinggian. Dengan data tersebut kita dapat menghitung ketinggian absolut dari suatu tempat pengukuran yang cocok dengan permukaan air laut menggunakan Δh, dengan $h_0 = 0$. Nilai tersebut merupakan nilai h_0, sehingga dapat diketahui perbedaan ketinggian seperti contoh di 2.5.3.1.

Pada metode ini, akurasi dari pengukuran tergantung dari stabilitas cuaca (tekanan barometric). Hal ini akan berubah selama kenaikan menuju h_1, dan perubahan ini harus dimasukan ke dalam perhitungan. Berikut seperti contoh di poin 2.5.3.1 nilai $p(h_0)$ harus selalu diperhatikan setiap sinkronisasi sensor tekanan pada lokasi h_0 dan jika ini dibutuhkan untuk mengukur ketinggian h_1.

Kita harus sadar bahwa memperkirakan akurasi itu penting, persamaan ini merupakan sebuah pendekatan. Dan yang paling dibutuhkan adalah perubahan linear temperature (tanpa memperhatikan perubahan adiabatic pada temperature, pencampuran udara yang disebabkan oleh perubahan cuaca, kelembaban, dsb, dan tetap menganggap M dan g adalah konstan) dan kondisi cuaca barometric yang stabil. Bagaimanapun juga ketidakakuratan persamaan ketika menentukan ketinggian, disebabkan oleh presisi atau tidaknya suatu sensor.

2.4.4 Sensor tekanan untuk altimeter

Kebanyakan altimeter yang ada di pasaran pada dasarnya merupakan sensor tekanan yang memiliki rentang dari -100 hingga 4000, 5000, atau 9000 meter debfab resolusi maksimal 14bit. Hal tersebut tidak bisa dikatakan bahwa sensor tersebut presisi. Hingga sekarang, mendapatkan sebuah resolusi yang lebih baik merupakan penelitian yang menghabiskan banyak dana.

Pada umumnya, akurasi juga sama pentingnya dengan resolusi untuk penggunaan luar ruangan, sifat temperature instrumentasi pengukuran. Faktor lapangan utama yang berkaitan dengan mobile system adalah ukurannya dan catu daya domestik. Semua karakter kombinasi ini terdapat pada MS5611 24bit sensor tekanan. Suatu miniatur sensor tekanan absolut yang digunakan untuk aplikasi dengan mobilitas tinggi. Untuk menentukan sebuah tekanan, sensor memiliki akurasi yang tinggi untuk mengukur suhu yang juga dibutuhkan untuk pengukuran ketinggian seperti persamaan 7. Dengan ukuran yang kecil dan kebutuhan daya yang minim, MS5611 sangat cocok untuk aplikasi yang membutuhkan mobilitas tinggi.

2.4.5 Sensor tekanan absolut MS5611

MS5611 merupakan sebuah teknologi semiconductor mutakhir. Komponen pada sensor tekanan absolut adalah sebuah elemen penginderaan silicon yang bekerja dengan prinsip piezoresistif. Elemen ini memiliki membran yang tipis yang peka secara anisotropis dari sebuah chip silicon. Hal itu merupakan sebuah poin yang cocok untuk sebuah atom local yang ditanamkan di kristal silicon, sehingga menciptakan sebuah zona dengan konduktivitas listrik yang berubah dengan memiliki sifat resistif. Ketika tekanan diaplikasikan pada elemen tersebut, membran tipis silicon akan terdeformasi. Gaya internal menghasilkan perubahan reversible dari struktur kristal. Khususnya di area resistor terdapat pergeseran potensial dalam struktur kristal yang mengarah pada perubahan terukur dalam nilai elektris yang disebut efek piezoresistif. Resistor dihubungkan dalam sebuah brige sehingga didapatkan tegangan listrik. Elemen sensor ini dipasangkan pada sebuah PCB dan secara elektrik terkoneksi dengan kawat tembaga seperti Gambar 2.7.

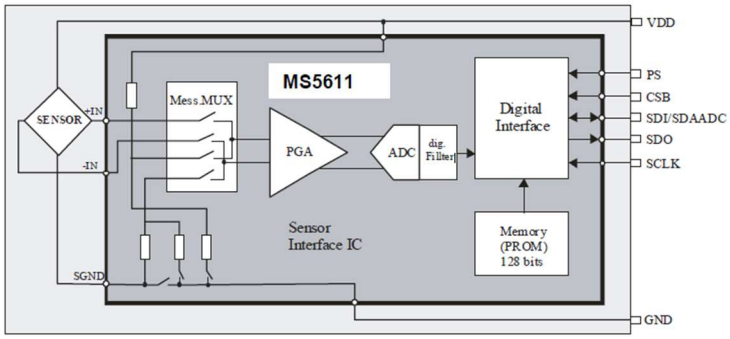
2.4.6 Sirkuit evaluasi sinyal

Disamping elemen sensor juga terdapat sirkuit terintegrasi dengan aplikasi spesifik atau ASIC yang prinsip fungsinya mirip dengan presisi ADC. Dimana ASIC akan merekam semua sinyal dan mengubah elemen sensor sinyal analog ke dalam 24 bit untuk nilai tekanan dan temperature. ASIC terdiri dari sebuah multiplexer, sebuah converter A/D sigma/delta 24bit, EPROM dan output digital dengan SPI dan I^2C *interface.*

Teknologi driver yang special (multiplexer) membuat sensor beroperasi dengan konsumsi daya rendah. Ini dapat terjadi dengan mengabaikan kenaikan temperature pada sisi sensitif sensor, menghasilkan stabilitas sinyal output pada elemen sensor.

Konverter mengukur keduanya baik voltase keluaran dari elemen sensor dan temperatur dari tahanan yang ada pada rangkaian bridge. Sinyal ini akan digunakan untuk mengkompensasi ketergantungan temperature pada sinyal tekanan dan menghasilkan sebuah peralatan thermometer dengan presisi tinggi (dengan resolusi 0.01°C). Dengan menggunakan rangkaian hambatan bridge untuk pengukuran temperature langsung (tanpa gradien temperature antara elemen sensor dan sensor temperatur), meskipun dengan perubahan temperatur metode pengkompensasian sensor memiliki efektifitas yang tinggi.

Dengan konverter A/D sigmal/delta 24bit yang sudah dioptimalkan, maka itu akan menghasilkan hubungan linear yang baik dan gangguan yang minim terhadap tegangan suplai dan rentang temperature. Berikut merupakan skema diagram cara kerja MS5611 [11].



Gambar 2.12 Sirkuit prinsip MS5611

2.4.7 Korelasi pada chip sensor

Bersama dengan SPI dan I^2C interface, AIS menawarkan interface antara elemen penginderaan tekanan piezoresistif dan sebuah kontroler luar menggunakan koefisisen kalibrasi pabrik. Koefisien koreksi disimpan pada EPROM 128bit selama proses manufaktur. Deviasi dari fungsi transfer ideal menyebabkan toleransi manufaktur, seperti pergerseran offset, variasi sensitifitas, ketidaklinearan, dan toleransi pada IC adalah sebuah pengukuran individualis dibawah kondisi tekanan

dan temperature dan dikonversikan dalam nilai korelasi. Pada pengoperasian, data koreksi sensor secara otomatis dibaca dari EPROM setelah catu daya direset.

2.5 Energi Gelombang Air Laut

2.5.1 Konsep energi gelombang air laut

Energi gelombang air laut merupakan sebuah akibat dari energi geombang angin. Energi ini banyak dimanfaatkan ntuk berbagai teknologi seperti pembangkit listrik, desalinasi air, dan untuk memompa air. Gelombang air laut dihasilkan oleh angin yang melewati permukaan laut. Selama gelombang air laut merambat lebih lambat dari kecepatan angin dan tepat diatas gelombang maka terjadi perpindahan energi dari angin ke gelombang. Perbedaan tekanan udara antara angin dan sisi lekukan gelombang, serta gesekan pada permukaan air oleh angin membuat air masuk ke tegangan geser dan menyebabkan munculnya gelombang. [8]

Tinggi gelombang ditentukan oleh kecepatan angin, durasi waktu angin bertiup, *fetch* (jarak antara tiupan angin dan gelombang), dan kedalaman serta topografi permukaan laut. Secara umum, gelombang yang lebih besar adalah gelombang yang lebih kuat yang ditentukan oleh cepat rambat gelombang, panjang gelombang dan massa jenis air.

Pergerakan naik turun (osilasi) dari gelombang tertinggi terdapat pada permukaan dan berkurang secara eksponensial berdasarkan kedalaman. Fluktuasi tekanan ini pada kedalaman yang lebih besar terlalu kecil.

Gelombang merambat dipermukaan laut, dan energi gelombang dan energi gelombang ditransportasikan oleh kecepatan. Kecepatan rata-rata laju transportasi dari gelombang pada bidang vertikal dalam satuan panjang sejajar dengan *wave crest* yang disebut dengan fluks atau energi gelombang per satuan meter (kW/m)

2.5.2 Rumus energi gelombang air laut

Di air yang dalam dimana kedalaman air lebih besar dari setengah panjang gelombang, fluks energi gelombang dirumuskan sebagai berikut:

$$P = \frac{\rho g^2}{64\pi} H^2 T \approx (0.5 \frac{\text{kW}}{m^3 s}) H^2 T \quad (7)$$

Dimana:

P = fluks energi gelombang (kW/m)

H = tinggi gelombang (m)

T = periode gelombang (s)

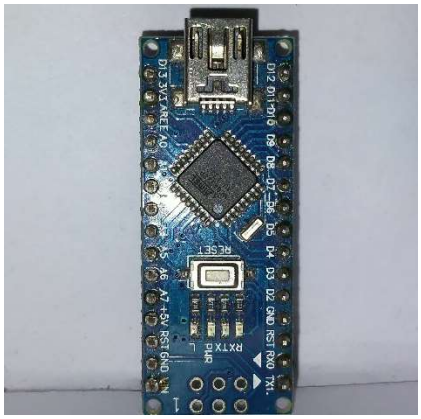
ρ = massa jenis air (kg/m^3)

g = percepatan gravitasi (m/s^2)

Dari rumus diatas dapat dilihat bahwa energi gelombang sebanding dengan periode (T) dan kuadrat tinggi gelombang (H^2). [9]

2.6 Komponen Alat Ukur

2.6.1 Arduino Nano



Gambar 2.2.13 Arduino Nano

Arduino merupakan sebuah platform dari *physical computing* yang bersifat *open source.* Arduino tidak hanya berupa hardware melainkan kombinasi antara hardware, Bahasa pemrograman dan *Integrated Developmnet Environment* (IDE). IDE merupakan sebuah software yang berperan untuk menulis program, meng-*compile* menjadi kode biner dan meng-*upload* ke dalam memori.

Arduino Nano adalah salah satu jenis dari produk Arduino yang berukuran kecil, lengkap dan mendukung penggunaan breadboard. Arduino nano dilengkapi dengan basis ic ATmega328 untuk versi 3.x dan ATmega 168 untuk versi 2.x.

2.6.1.1 Konfigurasi Pin Arduino Nano

- VCC merupakan pin yang berfungsi sebagai pin masukan catu daya.

- GND merupakan pin yang berfungsi sebagai ground

- AREF merupakan pin referensi tegangan input analog. Digunakan dengan fungsi analogReference().

- RESET merupakan pin yang digunakan untuk menghidupkan ulang Arduino. Biasanya dilengkapi dengan tombol reset pada papan utama Arduino.

- Serial RX (0) merupakan pin yang berfungsi sebagai penerima TTL data serial.

- Serial TX (1) merupakan pin yang berfungsi sebagai pengirim TTL data serial.

- External Interrupt (Interupsi Luar) merupakan pin yang dapat dikonfigurasi untuk memicu sebuah interupsi pada nilai yang rendah, meningkat atau menurun.

- Output PWM 8-Bit merupakan pin yang berfungsi untuk analogWrite()

- SPI merupakan pin yang berfungsi sebagai pendukung komunikasi

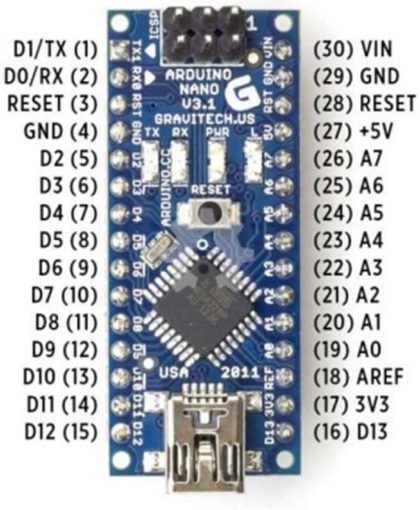
- LED berfungsi sebagai pin yang diset bernilai HIGH maka LED akan menyala dan ketika diset LOW maka LED akan padam.

- Input Analog (A0-A7) merupakan pin yang berfungsi sebagai pin yang dapat diukur/diatur mulai dari Ground sampai dengan 5V, juga memungkinkan untuk mengubah titik jangkauan tertinggi atau terendah mereka menggunakan fungsi analogReference().

Tabel 2.2 Konfigurasi Pin Arduino

Pin	Keterangan Pin
1	Digital Pin 1 (TX)
2	Digital Pin 0 (RX)
3 & 28	Reset
4 & 29	GND
5	Digital Pin 2
6	Digital Pin 3 (PWM)
7	Digital Pin 4
8	Digital Pin 5 (PWM)
9	Digital Pin 6 (PWM)
10	Digital Pin 7
11	Digital Pin 8
12	Digital Pin 9 (PWM)
13	Digital Pin 10 (PWM-SS)
14	Digitl Pin 11 (PWM-MOSI)
15	Digital Pin 12 (MISO)
16	Digital Pin 13 (SCK)
17	3V3
18	AREF

19	Analog Input 0
20	Analog Input 1
21	Analog Input 2
22	Analog Input 3
23	Analog Input 4
24	Analog Input 5
25	Analog Input 6
26	Analog Input 7
27	VCC
30	Vin



Gambar 2.14 Layout Konfigurasi Pin Arduino Nano

Sumber: repository.usu.ac.id

2.6.1.2 Spesifikasi Arduino Nano

Berikut ini adalah Spesifikasi yang dimiliki oleh Arduino Nano:

- MikrokontrolerAtmel ATmega168 atau ATmega328
- 5 V Tegangan Operasi
- 7-12VInput Voltage (disarankan)
- 6-20VInput Voltage (limit)
- Pin Digital I/O14 (6 pin digunakan sebagai output PWM)

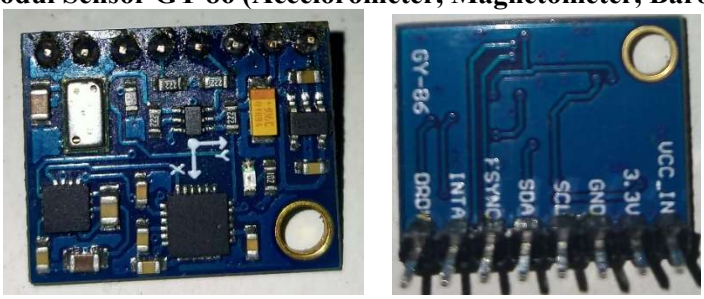
- 8 Pin Input Analog
- 40 mA Arus DC per pin I/O
- Flash Memory16KB (ATmega168) atau 32KB (ATmega328) 2KB digunakan oleh Bootloader
- 1 KbyteSRAM (ATmega168) atau 2 Kbyte(ATmega328)
- 512 ByteEEPROM (ATmega168) atau 1Kbyte (ATmega328)
- 16 MHz Clock Speed
- Ukuran1.85cm x 4.3cm

2.6.1.3 Catu Daya Arduino Nano

Arduino Nano dapat diaktifkan melalui koneksi USB Mini-B, atau melalui catu daya eksternal dengan tegangan belum teregulasi antara 6-20 Volt yang dihubungkan melalui pin 30 atau pin VIN, atau melalui catu daya eksternal dengan tegangan teregulasi 5 volt melalui pin 27 atau pin 5V. Sumber daya akan secara otomatis dipilih dari sumber tegangan yang lebih tinggi. Chip FTDI FT232L pada Arduino Nano akan aktif apabila memperoleh daya melalui USB, ketika Arduino Nano diberikan daya dari luar (Non-USB) maka Chip FTDI tidak aktif dan pin 3.3V pun tidak tersedia (tidak mengeluarkan tegangan), sedangkan 10 LED TX dan RX pun berkedip apabila pin digital 0 dan 1 berada pada posisi HIGH. [12]

2.6.2 Komponen Sensor

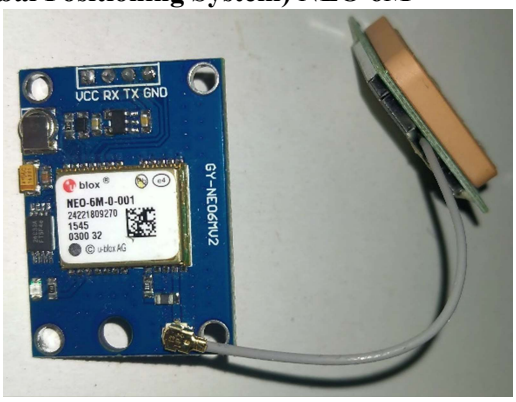
2.6.2.1 Modul Sensor GY 86 (Accelorometer, Magnetometer, Barometer)



Gambar 2.15 GY 86

Modul sensor GY 86 merupakan perangkat sensor gabungan dari beberapa sensor yang berupa sensor *3-axis accelerometer* dan *3-axis gyroscope* (MPU6050), *3-axis magnetometer* (HMC5883L), dan *barometer* (MS5611) serta dilengkapi *interface 12C.* Modul sensor ini membutuhkan suplai tegangan 3-5V dengan dimensi yang cukup kecil berukuran 21.5 x 16.8 x 2 mm.

2.6.2.2 GPS (Global Positioning System) NEO 6M

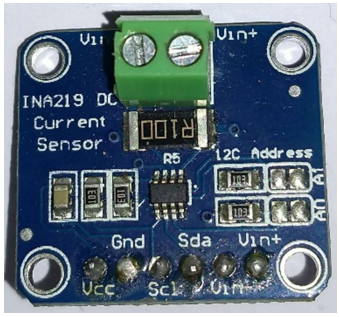


Gambar 2.16 GPS NEO 6M

Global Positioning System atau GPS merupakan peralatan yang mendeteksi lokasi pada permukaan bumi. Perangkat ini akan menerima koordinat dari satelit setiap detik dengan data waktu dan tanggal yang sesuai dengan UTC *(Universal Time Coordinated).*

2.6.2.3 Sensor Temperatur Air DS18B20

2.6.2.4 Sensor Arus INA219

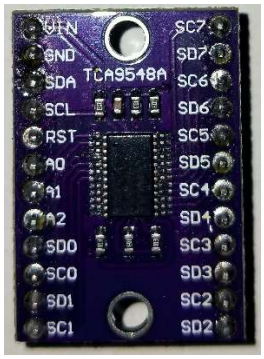


Gambar 2.17 Sensor Arus INA219

INA219 adalah module sensor DC yang dapat mengukur arus beban dan tegangan beban dengan presisi sampai 1%. Modul ini dapat mengukur arus beban sampai dengan 3.2A dan tegangan bus DC sampai dengan 26Volt. Koneksi modul ini adalah I2C sehingga sangat cocok untuk digunakan dengan Arduino, Raspberry Pi atau microcontroller sejenis yang mendukung koneksi I2C. Modul ini nanti akan digunakan untuk memonitoring daya dari baterai sebagai suplai daya utama perangkat alat ukur ketinggian gelombang.

2.6.3 Komponen Pengolah Data

2.6.3.1 Multiplexer I²C Adafruit TCA9548A

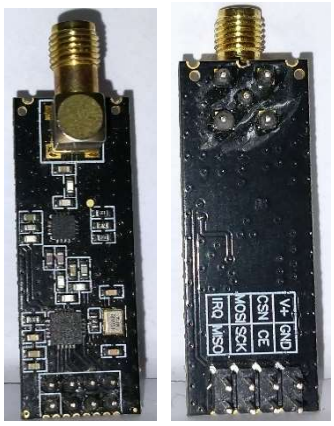


Gambar 2.18 *Multipexer* I²C Adafruit TCA9548A

Multiplexer adalah rangkaian logika kombinasional yang dirancang khusus untuk mengalihkan salah satu dari beberapa jalur INPUT (masukan) ke satu jalur OUTPUT (keluaran). Jalur Input yang terpilih menentukan input mana yang akan terhubung ke output. Multiplexer yang juga sering disingkat menjadi MUX atau MPX ini pada dasarnya berupa rangkaian digital yang dibuat dari gerbang logika berkecepatan tinggi yang digunakan untuk beralih data digital atau biner atau dapat berupa tipe analog yang menggunakan komponen transistor, MOSFET atau relay untuk mengalihkan salah satu input ke output. Multiplexer juga sering disebut juga sebagai Perangkat Pemilih Data (Data Selector). [13]

2.6.4 Komponen Pengirim Data

2.6.4.1 Radio Modul NRF24L01



Gambar 2.19 Modul Radio NRF24L01

Modul radio NRF24L01 merupakan sebuah modul komunikasi yang memanfaatkan pita gelombang RF 2.4GHz ISM *(Industrial, Scientific, and Medical)* dengan jangkauan akses 1.1 km. Modul ini menggunakan *interface* SPI untuk komunikasi

dan membutuhkan catu daya 5V untuk beroperasi. Modul ini memiliki 8 buah pin diantaranya adalah VCC, GND, CE, CSN, MOSI, MISO, SCK, dan IRQ.

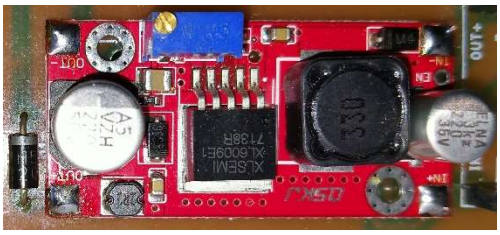
2.6.5 Komponen Sistem Catu Daya

2.6.5.1 Charge Controller TP4056 5V,1A

Gambar 2.20 *Charge Controller* TP4056

Charge controller atau pengatur pengisian daya berfungsi sebagai regulator pengisian daya dari power supply menuju baterai agar tetap menjaga baterai tidak *overcharging*. Karena spesifikasi baterai yang digunakan tidak boleh discharge lebih dari 4.2-volt. Spesifikasi modul ini adalah mampu mengatur tegangan keluaran menjadi 5-volt dengan arus 1A.

2.6.5.2 Boost Converter XL6009

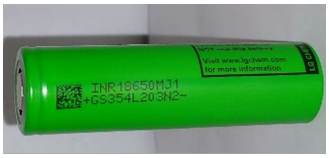


Gambar 2.21 *Boost Converter* XL6009

Modul XL6009 merupakan converter yang berfungsi untuk mengubah tegangan masukan menjadi tegangan keluaran yang lebih tinggi atau lebih rendah. Tegangan keluaran bisa diatur dengan cara memutar trimpot. Jika terjadi perubahan tegangan input, maka tegangan output akan tetap stabil karena modul ini juga berfungsi sebagai regulator. Spesifikasi modul ini adalah sebagai berikut:

- Input tegangan : 5-32V
- Output tegangan : 1.5-35V
- Tipe modul : *Non isolated step up* dan *step down*
- Penyearah : *Non-synchronous rectification*
- Frekuensi : 400kHz
- Arus keluaran rata-rata : 3 A
- Arus keluaran maksimal: 4 A

2.6.5.3 Baterai 18650 3.6 V 3400 mAH



Gambar 2.22 Baterai 18650 3.6V 3400 mAH

Salah satu jenis baterai yang banyak digunakan saat ini adalah *Lithium Battery* 18650. Ini adalah jenis battery yang dapat di cas ulang *(rechargeable).* Kebanyakan perangkat elektronik portable yang membutuhkan tenaga besar dan tahan lama dipastikan menggunakan battery 18650. Sebut saja misalnya laptop, power bank, wireless bluetooth speaker, perangkat remote control, mobil, helicopter. Alat pertukangan bor atau obeng wireless, lampu senter LED, rokok elektronik dan sebagainya.

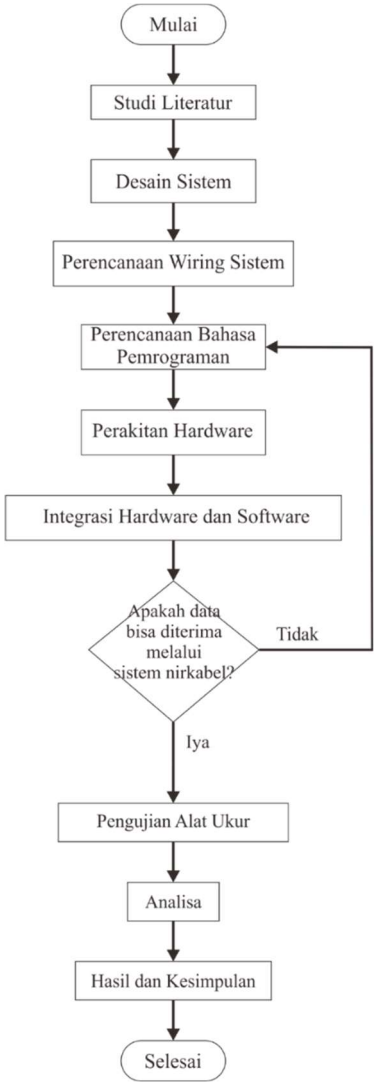
Nama battery 18650, merujuk pada ukuran fisiknya yang berbentuk silinder. Angka 18 untuk diameter battery 18 mm dan angka 650 untuk ukuran tinggi battery, 65,0 mm. Angka "0" dibelakang koma merujuk pada toleransi tinggi total battery berdasarkan jenis produk battery 18650 tersebut.

Tegangan kerja battery 18650 adalah 3,7 Volt. Maksimum dapat di cas 4,2 Volt dan battery kosong pada 3,0 Volt. Sedang kemampuan menyimpan arus listrik beragam tergantung produksinya. Dan secara umum diketahui battery ini maksimal meiliki kapasitas 3600 mAH. Ada yang 3400 mAH, 2500 mAH, 2200 mAH, 1500 mAH dan sebagainya. Namun maksimal yang dapat diproduksi hingga kini hanya dapat menyimpan arus maksimal 3600 mAH. [17]

BAB 3
METODOLOGI PENELITIAN

Dalam bab ini akan dijelaskan beberapa tahapan penelitian, dimulai dengan melakukan studi literatur hingga diperolehnya desain perangkat sistem untuk alat ukur tinggi gelombang air laut.

3.1 Langkah-Langkah Penelitian



Gambar 3.1 Diagram Alir Penelitian

3.1.1 Identifikasi dan Perumusan Masalah

Pada sub bab ini dilakuan perumusan masalah awal dengan cara melakukan pencarian informasi terkait alat ukur ketinggian gelombang air laut. Pencarian informasi awal

didapatkan dari sumber-sumber melalui pencarian dari data sekunder. Setelah itu penulis dapat menentukan perumusan masalah yang terkait tujuan penelitian.

3.1.2 Studi Literatur

Studi literatur dilakukan dengan tujuan untuk medapatkan konsep serta metode yang berhubungan dengan pemecahan masalah dan solusi dalam melakukan penelitian ini. Penelitian ini menggunakan studi literatur sebagai data sekunder untuk acuan penelitian. Pada penelitian ini, studi literatur yang digunakan untuk melakukan penelitian adalah penelitian terdahulu. Studi literatur yang digunakan yaitu penelitian tentang alat ukur ketinggian air laut dan alat ukur deteksi tsunami.

3.1.3 Desain Sistem

Pada sub bab ini akan dilakukan proses desain sistem sensor, transfer data dan catu daya dari alat ukur ketinggian air laut. Pada proses desain akan dijelaskan bagaimana konfigurasi pin peralatan terhadap mikrokontroler Arduino Nano.

3.1.4 Perencanaan Wiring Sistem

Perencanaan wiring sistem dimaksudkan adalah perencanaan bagaimana semua komponen akan dijadikan satu kesatuan. Perencanaan wiring ini dilakukan dengan menggunakan software Eagle. Sehingga hasil akhir dari desain wiring alat ukur ini adalah desain layout PCB.

3.1.5 Perencanaan Bahasa Pemrograman

Perencanaan Bahasa pemrograman dimaksudkan untuk mengaktifkan komponen sistem yang meliputi:

a. Sistem penginderaan (gy 86, gps, sensor air, dan sensor arus serta tegangan pada baterai)

b. Sistem penyimpanan dan pengiriman data

3.1.6 Perakitan Hardware

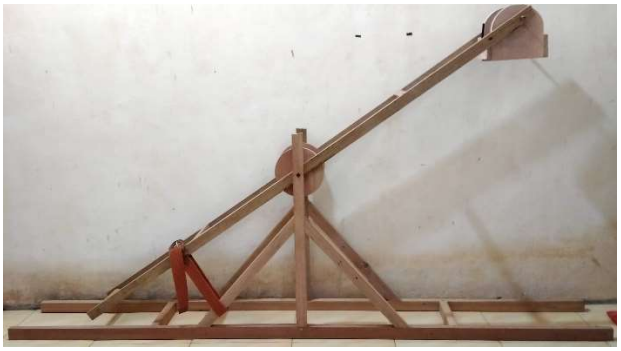
Setelah desain wiring sesuai dan alat bisa berfungsi ketika dirangkai pada breadboard. Maka PCB akan dicetak dan semua komponen akan diletakkan pada PCB dengan menggunakan pin header.

3.1.7 Integrasi Hardware dan Software

Integrasi hadware dan software diperlukan untuk menghubungkan antara perangkat alat ukur dengan software pengolah dan penampil data yang ada di darat. Data akan dikirimkan melalui gelombang radio dengan jangkauan jarak 1.1 km dan diterima oleh radio receiver yang sudah dihubungkan pada perangkat komputer sehingga akan ditampilkan berbagai jenis data diantara lain adalah data lokasi, data gelombang (tinggi, periode, serta daya), data temperature (air dan udara), dan data arus serta tegangan baterai.

3.1.8 Pengujian Alat Ukur

Setelah proses integrasi selesai maka langkah selanjutnya adalah melakukan uji coba dengan perangkat sederhana. Uji coba ini dilakukan untuk melihat bagaimana hasil pengukuran yang dilakukan alat uji.



Gambar 3.2 Simulator *Wave Buoy*

Dengan spesifikasi alat simulator *wave buoy*, sebagai berikut:

Tinggi ayun maximal = 110 cm

Panjang lengan ayun = 210 cm

3.1.9 Analisa Data

Dalam analisa data maka akan dibahas mengenai hasil pengukuran alat yang sudah diujikan dengan menggunakan alat simulator sederhana. Dengan alat simulator sederhana maka akan diketahui berapa error hasil pengukuran dari data gelombang.

3.1.10 Hasil dan Kesimpulan

Pada hasil dan kesimpulan akan disampaikan mengenai error yang terjadi pada alat ukur dan beberapa saran yang didapat dari kendala selama pengerjaan alat ukur ketinggian air laut.

3.2 Timeline Penelitian

Tabel 3.1 Timline pengerjaan tugas akhir

	Januari				Februari				Maret				April				Mei				Juni				Juli			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Identifikasi Masalah																												
Studi Literatur																												
Seminar Proposal																												
Desain Sistem																												
Perencanaan Wiring Sistem																												
Perencanaan Bahasa Pemrograman																												
Perakitan Hardware																												
Integrasi Hardware dan Software																												
Pengujian Alat Ukur																												
Analisa Data																												
Pengerjaan Laporan																												
Sidang Akhir																												

BAB 4 RANCANG BANGUN MODEL DAN PEMBAHASAN

4.1 Gambaran Umum

Pada bab ini akan dibahas secara detail bagaimana proses pembuatan alat ukur ketinggian air laut menggunakan modul gy 86 dengan metode perubahan tekanan relatif. Tahapan yang akan dibahas pada bab ini mulai dari proses desain sistem alat ukur, proses perencanaan wiring sistem, proses perencanaan bahasa pemrograman, proses perakitan hardware, proses integrasi software dan hardware, proses pengujian alat ukur, analisa data hasil pengukuran serta biaya pembuatan sistem elektronik alat ukur.

4.2 Desain Sistem Alat Ukur Ketinggian Air Laut

Proses desain sistem alat ukur ketinggian air laut *(wave buoy)*, akan dibagi menjadi beberapa subsistem yang terdiri dari sistem peginderaan, sistem penyimpanan serta pengiriman data, dan sistem catu daya. Secara keseluruhan desain sistem alat ukur dapat dilihat di dalam diagram blok yang ada pada Gambar 4.1.

4.2.1 Sistem penginderaan

Sistem penginderaan pada wave buoy terdiri dari penginderaan lokasi (gps), penginderaan tekanan untuk data gelombang air laut (ms5611), penginderaan temperatur air dan penginderaan arus serta tegangan baterai.

4.2.1.1 Penginderaan lokasi

Penginderaan lokasi pada *wave buoy* menggunakan perangkat sensor gps dengan merk Tiny GPS Neo 6M. Penginderaan lokasi ini nantinya akan diperoleh data koordinat titik dimana alat ukur diletakkan. Perangkat ini akan diaktifkan dengan menghubungkan pin rx gps ke pin 3 dan tx gps ke pin 4 Arduino Nano.

4.2.1.2 Penginderaan tekanan relatif

Penginderaan tekanan digunakan untuk mendapatkan data tekanan relative terhadap tekanan atmosfir. Tekanan ini nantinya akan dikonversikan menjadi ketinggian dengan menggunakan rumus (4). Dengan data dari tekanan yang dirubah menjadi ketinggian ini nanti akan diproses melalui algoritma sederhana menjadi data gelombang berupa ketinggian, periode, serta daya gelombang.

Penginderaan ini menggunakan sensor piezoelektrik jenis MS5611 yang sudah terpasang pada modul GY 86. MS5611 merupakan sensor barometer yang bisa mengukur tekanan dan juga temperatur udara. Sensor ini akan diaktifkan dengan menghubungkan pin scl ke pin scl 5 dan sda ke pin sda 5 pada perangkat *multiplexer* I^2C.

- Ketinggian gelombang

Ketika alat ukur mengalami pergerakan naik turun (osilasi) sensor tekanan akan mendeteksi ketinggian saat titik terendah osilasi dan titik tertinggi osilasi, selisih antara keduanya merupakan jarak antara puncak dan lembah gelombang. Sehingga untuk mendapatkan ketinggian gelombang, hasil selisih tersebut dibagi 2

- Periode gelombang

Ketika alat ukur berosilasi dan mencapai titik dimana lebih rendah dari setengah jarak puncak dan lembah maka perhitungan waktu akan dimulai, dan ketika pergerakan *wave buoy* kembali mencapai titik tersebut kembali maka perhitungan waktu akan berhenti. Hasil dari perhitungan waktu tersebut masih merupakan waktu rambat ½ gelombang sehingga harus dikalikan 2 untuk mendapatkan nilai periode gelombang.

- Daya gelombang

Setelah tinggi dan periode gelombang dapat diketahui maka daya gelombang didapat dengan mengkalikan tinggi gelombang pangkat 2 dengan periode gelombang sesuai dengan rumus (2).

4.2.1.3 Penginderaan temperature air

Penginderaan temperature air menggunakan sensor DS18B20. Sensor merupakan jenis sensor RTD dengan menggunakan system 1 kabel (onewire) untuk mengirimkan data penginderaan ke mikrokontroler. Kabel ini nanti akan dihubungkan melalui pin 5 Arduino Nano.

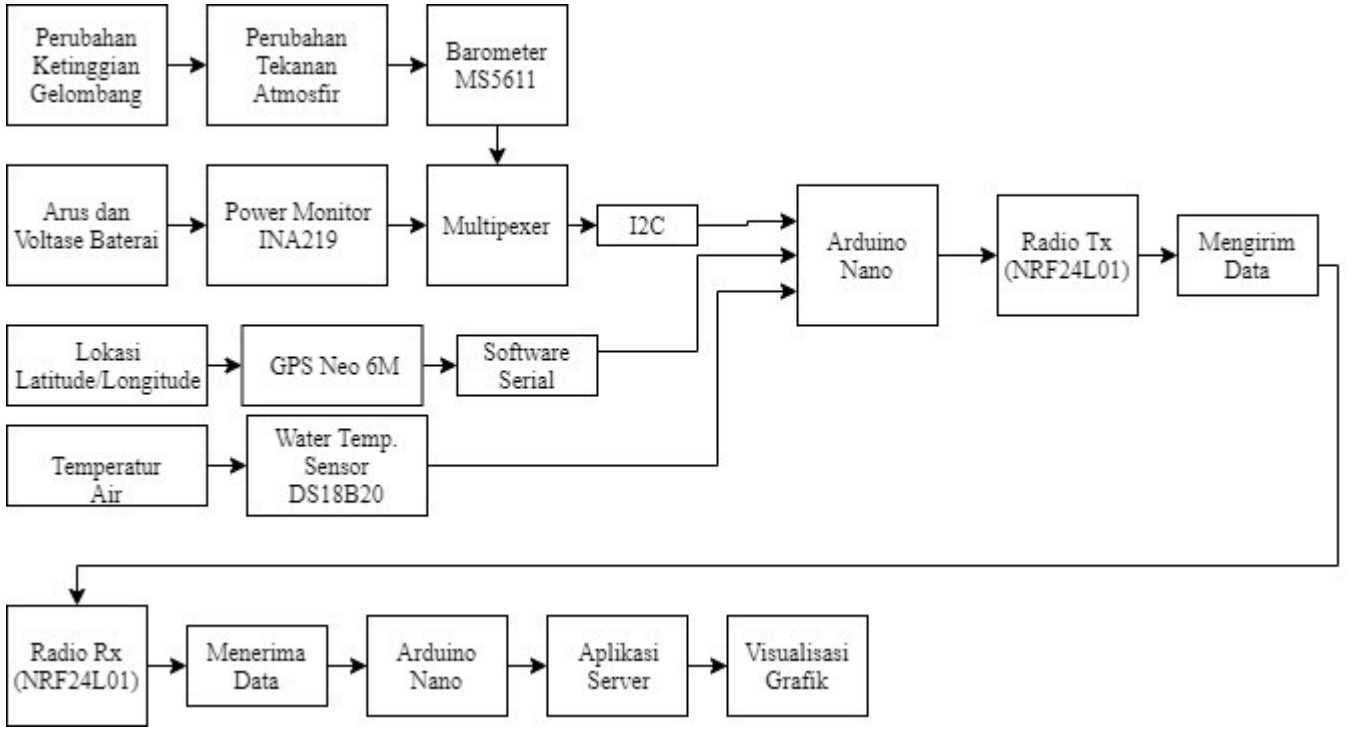
4.2.2 Sistem Pengiriman Data

4.2.2.1 Sistem Pengiriman Data

Sistem pengiriman data menggunakan modul radio jenis NRF24L01. Modul radio ini memiliki jangkauan maksimal 1.1 km. Untuk data yang ditransfer menggunakan modul radio adalah berupa data lokasi, data gelombang, data arus serta tegangan baterai dan data temperatur air. Tetapi dalam sistem transfer data melalui modul radio ini tidak dikirimkan data waktu dan tanggal. Modul radio akan diaktifkan dengan menghubungkan 7 pin modul ke Arduino Nano sebagai berikut;

Tabel 4.1 Konfigurasi NRF24L01 untuk Arduino Nano

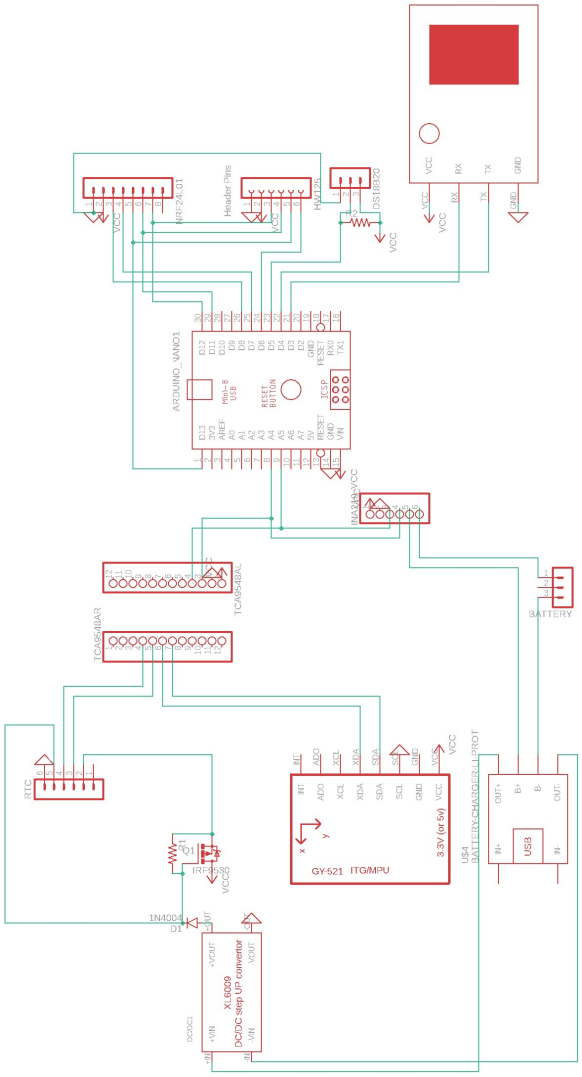
Pin NRF24L01	Pin Arduino Nano
CE	8
CSN	7
SCK	13
MISO	12
MOSI	11
VCC	3.3v



Gambar 4.1 Blok Diagram Desain Alat Ukur Ketinggian Air Laut

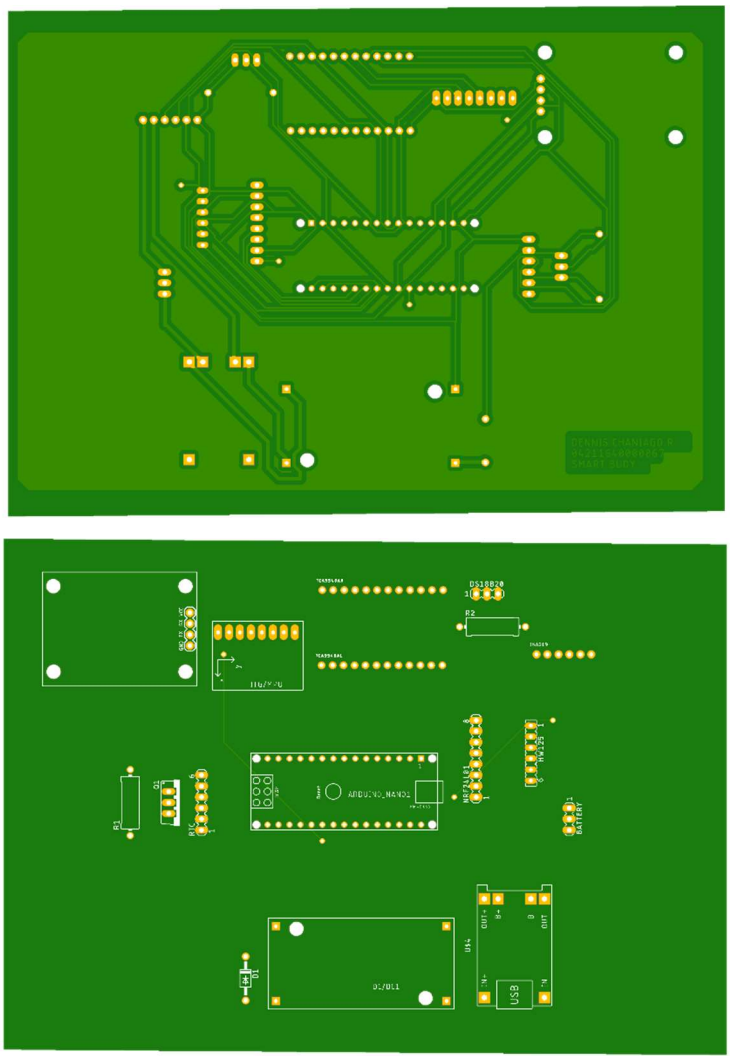
4.3 Perencanaan Wiring Sistem

Dalam merencanakan wiring sistem dari alat ukur digunakan sebuah software Eagle. Pertama adalah dengan membuat skematik sistem alat ukur, hasilnya adalah seperti Gambar 4.1



Gambar 4.2 Wiring skematik sistem alat ukur ketinggian air laut meggunakan software Eagle

Setelah dibuat skematik sistem wiring alat ukur maka langkah selanjutnya adalah membuat wiring yang sudah diplotkan dengan PCB. Hasil skematik yang sudah dibuat desain PCB nya dapat dilihat pada Gambar 4.2



Gambar 4.3 Atas: layout pcb bagian bawah; Bawah: layout pcb bagian atas

4.4 Perencanaan Bahasa Pemrograman

Dalam perencanaan bahasa pemrograman akan dibagi menjadi beberapa program diantaranya adalah:

- Program induk
- Program untuk aktivasi sensor gy 86
- Program untuk aktivasi gps
- Program untuk pengiriman data (radio)
- Program untuk manajemen konsumsi daya baterai (alarm)
- Program untuk aktivasi sensor temperatur air

4.4.1 Program Induk

Pada program induk akan dilakukan pengenalan semua komponen yang akan digunakan. Pengenalan dilakukan dengan cara mendeklarasikan library yang akan digunakan dengan menggunakan kode #include, sehingga program yang akan digunakan adalah seperti berikut:

```
#include <Wire.h>
#include <SPI.h>
#include <SD.h>
#include <TinyGPS++.h>
#include <SoftwareSerial.h>
#include <I2Cdev.h>
#include <MPU6050.h>
#include <MS5611.h>
#include <DS3232RTC.h>
#include <Adafruit_INA219.h>
#include <RH_NRF24.h>
#include <inttypes.h>
#include <OneWire.h>
#include <DallasTemperature.h>
```

Selanjutnya adalah membuat program yang digunakan untuk mengenali paket-paket data yang akan disimpan maupun dikirimkan melalui radio. Dikarenakan data yang dikirimkan sangat banyak maka data dibuat beberapa paket. Paket data terdiri dari

beberapa val. Data yang akan dikirim dibuat menjadi 2 paket. Dengan setiap paket terdiri dari 5 val. Paket data pertama berisi val gps *latitude*, val gps *longitude,* val arus baterai, val tegangan baterai, dan val 5 diisi data nol. Diisi data nol agar sistem mengenali bahwa paket ini merupakan paket 1. Selanjutnya untuk paket 2 berisi data val ketinggian gelombang, val periode gelombang, val daya gelombang, val temperature udara, dan val temperatur air.

```
struct data_holder {
  float val1;
  float val2;
  float val3;
  float val4;
  float val5;
} packet1, packet2;

#define gpsLat packet1.val1
#define gpsLong packet1.val2
#define sCur packet1.val3
#define sVolt packet1.val4

}
```

```
#define waveHeight packet2.val1
#define wavePeriod packet2.val2
#define wavePower packet2.val3
#define airTemp packet2.val4
#define waterTemp packet2.val5

float gpsDate, gpsTime;

#define TCAADDR 0x70
void tcaselect(uint8_t i) {
  if (i > 7) return;
  Wire.beginTransmission(TCAADDR);
  Wire.write(1 << i);
  Wire.endTransmission();
```

Digunakan kode #define agar *compiler* mengerti bahwa kata awal akan diganti dengan kata akhir, semisal #define gpsLat packet1.val11 maka artinya adalah mengganti kata gpsLat dengan packet1.val11. Dalam program diatas disebutkan juga #define TCAADDR 0x70 untuk mengenali multiplexer yang digunakan.

Selanjutnya adalah memprogram void setup, void setup ini digunakan untuk mengenali program mana yang akan ditampilkan atau dikirim datanya. Dikarenakan program induk ini merupakan program utama yang selanjutnya akan ada program lain. Pengenalan program melalui void setup penting untuk mengetahui program mana yang mengalami error atau gagal dalam mengirim data.

```
void setup() {
  Serial.begin(9600);
  Wire.begin();

  packet1.val5=0;
  Serial.println("setup gps");
  setup_gps();
  setup_radio();
  Serial.println("gy");
  setup_gy86();
  Serial.println("power");
  setup_power_control();
  Serial.println("setup temperatur");
  setup_water_temperature();
  setup_logger();
}
```

Selanjutnya adalah program void loop yang digunakan untuk menampilkan data mana yang akan dicetak atau biasa disebut Serial.println.

```
void loop() {
  Serial.println("get gps data");
  get_gps_data();
  Serial.println("get wave");
  get_wave_stats();
  Serial.println("get water");
  get_water_temperature();
  Serial.println("set alarm");
  set_new_alarm();
  send_data();
  logger_save();
```

```
Serial.print(packet1.val1);
  Serial.print('\t');
  Serial.print(packet1.val2);
  Serial.print('\t');
  Serial.print(packet1.val3);
  Serial.print('\t');
  Serial.print(packet1.val4);
  Serial.print('\t');
  Serial.print(packet2.val1);
  Serial.print('\t');
  Serial.print(packet2.val2);
  Serial.print('\t');
  Serial.print(packet2.val3);
  Serial.print('\t');
  Serial.print(packet2.val4);
  Serial.print('\t');
  Serial.print(packet2.val5);
  Serial.print('\t');
  Serial.println();
//  turn_off();
}
```

4.4.2 Program Untuk Aktivasi Sensor Gy 86

Aktivasi sensor pada GY 86 yang dimaksud adalah sensor MS5611 yang merupakan barometer. Pemrograman ini akan dibuat agar sensor ms5611 dapat mengukur ketinggian, periode, serta daya gelombang.

Pertama adalah menginisiasi sensor ms5611 yang merupakan sensor barometer. Proses inisiasi juga melibatkan sensor mpu6050 yang merupakan sensor utama pada modul gy 86.

```
MPU6050 mpu;
MS5611 baro;
void setup_gy86(){
  Serial.println("tca select");
  tcaselect(5);
  Serial.println("set i2c");

mpu.setI2CMasterModeEnabled(false
);
  mpu.setI2CBypassEnabled(true) ;
  mpu.setSleepEnabled(false);
  Serial.println("initialize");
  mpu.initialize();
  Serial.println("baro begin");

baro.begin(MS5611_ULTRA_HIGH_
RES);
}
```

Untuk selanjutnya adalah membuat program untuk sensor ms5611 agar dapat mengukur ketinggian, periode, daya gelombang serta temperature udara.

```
void get_wave_stats(){
  tcaselect(5);
  long pressure;
  double altitude, min_height, max_height, wave_height, mid_point, smudge_factor, wave_power;
  byte escaped, started;
  unsigned long period_start, period_end;
  float avg_period = -1;

  unsigned long start_time = millis();

  airTemp = baro.readTemperature(true);

  pressure = baro.readPressure(true);
  altitude = baro.getAltitude(pressure);
  max_height = altitude;
  min_height = altitude;
```

Setelah beberapa definisi dikenali oleh sensor maka dibuat program untuk mengukur ketinggian gelombang.

```
while(millis() - start_time < 5000){
    pressure = baro.readPressure(true);
    altitude = baro.getAltitude(pressure);
    if (altitude < min_height) min_height = altitude;
    if (altitude > max_height) max_height = altitude;
  }
  mid_point = (max_height + min_height)/2.0;
  wave_height = (max_height - mid_point) / 2.0;

seaLevelPressure = baro.getSeaLevel(pressure, altitude);
  seaLevelAltitude = baro.getAltitude(seaLevelPressure);
  waveHeight = altitude - seaLevelAltitude;
```

Setelah ketinggian maka program selanjutnya untuk menghitung periode gelombang.

```
smudge_factor = wave_height * 0.15;
start_time = millis();
  while(millis() - start_time < 5000){
    pressure = baro.readPressure(true);
    altitude = baro.getAltitude(pressure);
    if (altitude < mid_point + smudge_factor && altitude > mid_point - smudge_factor){
      if ( !started){
        period_start = millis();
        started = true;
      }
      else{
        if ( escaped ){
          period_end = millis();
          started = false;
          escaped = false;
          if(avg_period != -1){
            avg_period = (avg_period + (period_end-period_start)*2)  / 2.0;
          }
          else{
            avg_period =  (period_end-period_start)*2;
          }
        }
      }
    }
    else{
      escaped = true;
    }
  }
    wavePeriod = avg_period/1000;
```

Setelah ketinggian dan periode gelombang didapat maka daya gelombang dapat dihitung dengan program berikut.

```
  wavePower = int(0.5 * wave_height * wave_height * avg_period / 1000);

}
```

4.4.3 Program Untuk Aktivasi GPS

Langkah pertama adalah dengan mengenalkan perangkat gps ditempatkan pada pin 3 dan 4 mikrokontrol.

```
TinyGPSPlus gps;
// Arduino -> GPS
// d4->tx, d3->rx
SoftwareSerial ss(4, 3);
void setup_gps(){
  ss.begin(9600);
}
```

Setelah itu program untuk mendapatkan data koordinat lokasi.

```
void get_gps_data(){
  unsigned long start_time = millis();

  while ( millis()-start_time < 300 ) {
    while (ss.available() > 0) {
      if (gps.encode(ss.read())) {
        if (gps.location.isValid()){
           gpsLat = gps.location.lat();
           gpsLong = gps.location.lng();
        }
        if (gps.date.isValid() && gps.time.isValid()) {
           gpsDate = gps.date.value();
           gpsTime = gps.date.value();
        }
      }
    }
    delay(10);
    if( gps.location.isValid() && gps.date.isValid() &&
gps.time.isValid() ){
      break;
    }
  }
}
```

4.4.4 Program Pengiriman Data Melalui Radio

4.4.4.1 Perangkat pengirim *(Transmitter)*

Langkah pertama adalah mengenalkan perangkat radio terletak pada pin 8 dan 7 dan membuat alamat radio pengirim sesuai dengan penerima.

```
RH_NRF24 nrf24(8, 7);

void setup_radio(){
  Serial.begin(9600);
  if (!nrf24.init())
    Serial.println("init failed");
  // Defaults after init are 2.402 GHz (channel 2), 2Mbps, 0dBm
  if (!nrf24.setChannel(1))
    Serial.println("setChannel failed");
  if                                    (!nrf24.setRF(RH_NRF24::DataRate2Mbps,
RH_NRF24::TransmitPower0dBm))
    Serial.println("setRF failed");
}
```

Lalu perintahkan radio untuk mengirim paket data.

```
byte sz = sizeof(packet1);
byte buf[sizeof(packet1)] = {0};

void send_data(){
  memcpy(buf, &packet1, sz);
  nrf24.send(buf, sz);
  nrf24.waitPacketSent();
  delay(300);

  memcpy(buf, &packet2, sz);
  nrf24.send(buf, sz);
  nrf24.waitPacketSent();
  delay(300);
}
```

4.4.4.2 Perangkat penerima *(Receiver)*

Pertama dibuat program untuk mengenali perangkat radio penerima pada mikrokontroler.

```
#include <ArduinoJson.h>
#include <SPI.h>
#include <RH_NRF24.h>

RH_NRF24 nrf24(8, 7);
```

Selanjutnya adalah menyiapkan *struct data* yang harus sesuai dengan program pengirim.

```
struct data_holder {
  float val1;
  float val2;
  float val3;
  float val4;
  float val5;
} packet;
```

Setelah itu mengenalkan istilah data yang akan diterima dengan kode #define

```
#define gpsLat packet.val1
#define gpsLong packet.val2
#define sCur packet.val3
#define sVolt packet.val4
#define isPacket1 packet.val5

#define waveHeight packet.val1
#define wavePeriod packet.val2
#define wavePower packet.val3
#define airTemp packet.val4
#define waterTemp packet.val5
```

Lalu program untuk penerima data yang berupa *byte* dan akan dirubah kembali menjadi *struct*, data ini nantinya akan disimpan dalam json. Jika data akan dibuka maka perlu di*serealize* terlebih dahulu. Tetapi data ini akan banyak memakan memori dari mikrokontroler, maka jika data tidak digunakan akan langsung dibuang atau *garbagecollect*.

```
StaticJsonDocument<1024> doc;

bool isPacket1Received = false;

void setup()
{
  Serial.begin(9600);
  if (!nrf24.init())
    Serial.println("init failed");
  // Defaults after init are 2.402 GHz
(channel 2), 2Mbps, 0dBm
  if (!nrf24.setChannel(1))
    Serial.println("setChannel failed");
  if
(!nrf24.setRF(RH_NRF24::DataRate2
Mbps,
RH_NRF24::TransmitPower0dBm))
    Serial.println("setRF failed");
}

void loop()
{
  if (nrf24.available())
  {
    uint8_t buf[sizeof(packet)];
    uint8_t len = sizeof(buf);
    if (nrf24.recv(buf, &len))
    {
      memcpy(&packet,                buf,
sizeof(packet));
```

```
      if(isPacket1 == 0){
        isPacket1Received = true;
        doc["Latitude"] = gpsLat;
        doc["Longitude"] = gpsLong;
        doc["Current"] = sCur;
        doc["Voltage"] = sVolt;
      }
      if(isPacket1       !=       0       &&
isPacket1Received == true){
        doc["WaveHeight"]                =
waveHeight;
        doc["WavePeriod"]                =
wavePeriod;
        doc["WavePower"]                 =
wavePower;
        doc["AirTemp"] = airTemp;
        doc["WaterTemp"] = waterTemp;
        serializeJson(doc, Serial);
        Serial.println();

        isPacket1Received = false;
        doc.garbageCollect();
      }
      if(isPacket1       !=       0       &&
isPacket1Received == false){
        doc.garbageCollect();
      }
    }
  }
}
```

4.4.5 Program Manajemen Daya Baterai

Pada program ini akan diatur waktu pengisian dengan men-*cut off* baterai untuk mensuplai ke sistem dengan menggunakan MOSFET. Sehingga ketika daya baterai rendah maka baterai akan di-*charging* terlebih dahulu.

```
Adafruit_INA219 ina219;

void setup_power_control(){
  ina219.begin();
  ina219.setCalibration_32V_1A();
}
void set_new_alarm(){
  int wait_time = get_wait_time_from_voltage();

  tcaselect(4);
  RTC.alarmInterrupt(ALARM_1, true);
  RTC.squareWave(SQWAVE_NONE);
  setTime(0, 0, 0, 1, 1, 1970);
  RTC.set(now());

  // set new alarm
  RTC.setAlarm(ALM1_MATCH_MINUTES, 0, wait_time, 0, 1);
}
int get_wait_time_from_voltage(){
  float shuntvoltage = ina219.getShuntVoltage_mV();
  float busvoltage = ina219.getBusVoltage_V();
  float loadvoltage = busvoltage + (shuntvoltage / 1000);
  sCur = ina219.getCurrent_mA();
  sVolt = loadvoltage;

  // Samsung 18650 % capacity at a given voltage
  // batt_voltages          0.0, 3.4, 3.5, 3.6, 3.7, 3.8, 3.9, 4.0, 4.1, 4.2, 4.5
  // batt_percentages         0,   0,   9,  22,  52,  64,  75,  84,  93, 100, 100
  int wait_time;
  if (loadvoltage < 3.6) wait_time = 15;
  else if (loadvoltage < 3.8) wait_time = 10;
  else if (loadvoltage < 4.0) wait_time = 6;
  else if (loadvoltage < 4.1) wait_time = 3;
  else wait_time = 2;

  return wait_time;
}
void turn_off(){
  // clear old alarm flag - turning off system
  RTC.alarm(ALARM_1);
}
```

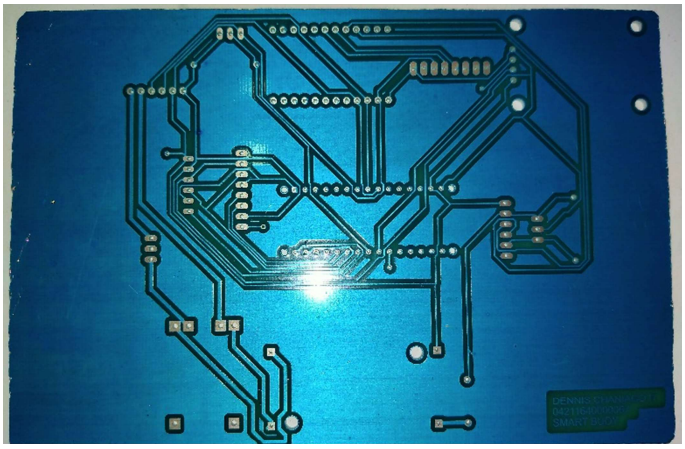
4.4.6 Program untuk Sensor Temperatur Air

Mengukur temperatur air dengan menggunakan rtd jenis ds18b20 dengan sistem satu kabel atau *one wire* maka dibutuhkan program sebagai berikut:

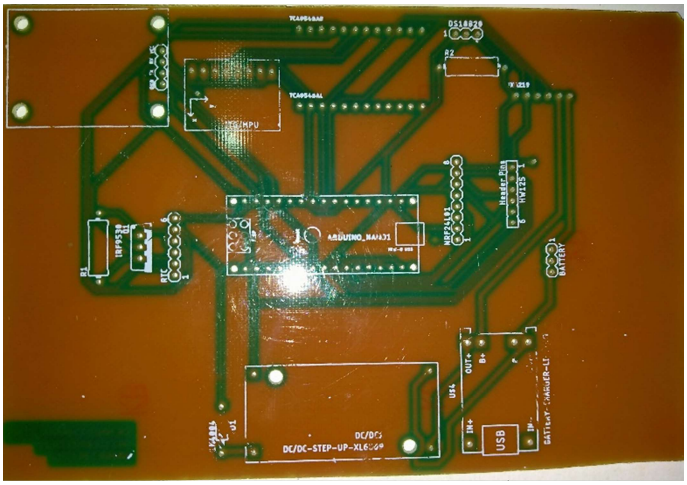
```
OneWire oneWire(5);
DallasTemperature
sensors(&oneWire);


void setup_water_temperature(){
  sensors.begin();
}

void get_water_temperature(){
  sensors.requestTemperatures();
  waterTemp                          =
sensors.getTempCByIndex(0);
}
```

4.5 Perakitan Hardware

Proses perakitan hardware merupakan proses untuk menenempatkan semua komponen elektronik ke dalam PCB yang sudah dicetak. Jenis PCB yang digunakan adalah PCB Single Layer dengan *masking* biru. Berikut bentuk PCB untuk *Smart Buoy Client* yang sudah jadi dapat dilihat pada Gambar 4.3 dan 4.4.

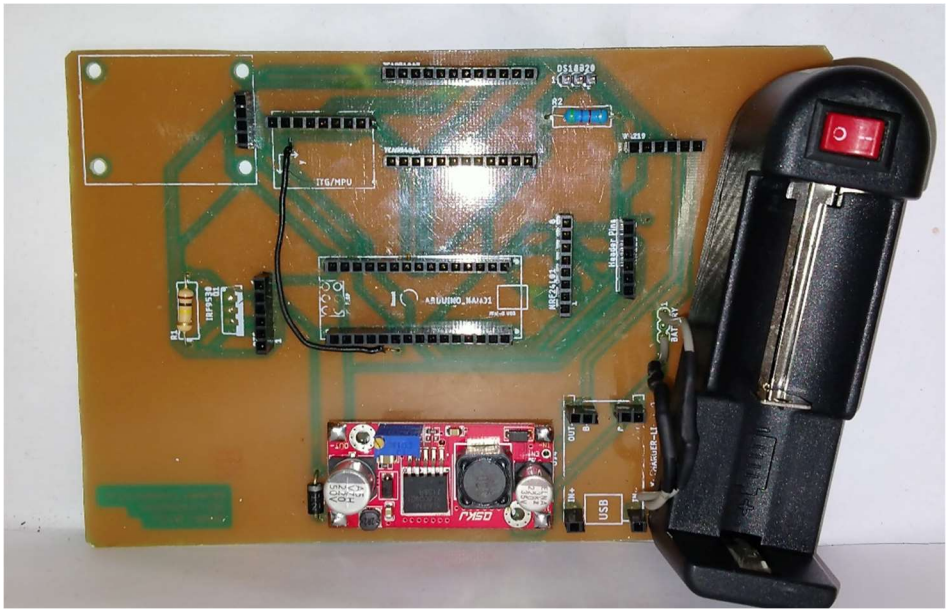


Gambar 4.4 PCB tampak bawah



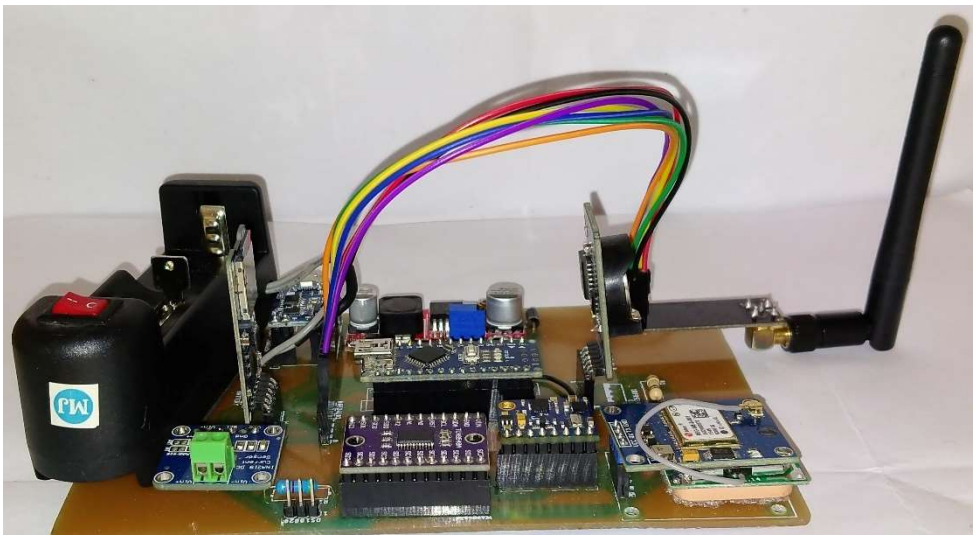
Gambar 4.5 PCB tampak atas

Selanjutnya adalah memasang pin header pada PCB dan beberapa komponen yang bersifat tidak lepas pasang seperti resistor, tempat battery, dan dc-dc converter. Pemberian pin *header* dimaksudkan agar komponen dapat dengan mudah dilepas jika terjadi kerusakan sehingga memudahkan proses reparasi maupun pengecekan. Bentuk PCB yang sudah dilengkapi dengan pin header dapat dilihat pada Gambar 4.5.



Gambar 4.6 PCB dengan pin header

Setelah pin *header* semua terpasang maka langkah selanjutnya adalah memasang semua komponen elektronik pada PCB. Berikut PCB yang sudah dilengkapi dengan semua komponen dapat dilihat pada Gambar 4.6.



Gambar 4.7 Perangkat Pengirim *(Smart Buoy Client)*

Pada Gambar 4.6 merupakan perangkat pengirim data *(Client),* sedangkan untuk perangkat penerima *(Server)* tidak menggunakan PCB yang sudah didesain dengan jalurnya melainkan dihubungkan dengan kabel pada PCB *holes.* Perangkat penerima *(Smart Buoy Server)* dapat dilihat pada Gambar 4.7.

Gambar 4.8 Perangkat Penerima *(Smart Buoy Server)*

4.6 Integrasi Software dan Hardware

Integrasi *software* dan *hardware* menggunakan software Visual Studio Community 2019. Integrasi ini bertujuan untuk mempermudah tampilan dari data yang dikirim oleh *client* menuju *server.* Selain ditampilkan dalam bentuk grafik nanti data yang dikirim akan disimpan dalam bentuk Microsoft Acces.

Berikut langkah bagaimana mengintegrasikan *software* dengan *hardware* menggunakan Visual Studio:

4.6.1 Membuat Program Tampilan *Dashboard*

Program untuk membuat tampilan *dashboard* atau Main.cs merupakan sebuah program untuk mengatur tata letak serta tampilan dari aplikasi server.

4.6.1.1 Menyiapkan variabel dan objek aplikasi

Langkah pertama adalah menyiapkan semua variabel dan objek yang dibutuhkan untuk menjalankan aplikasi server. Untuk memudahkan memahami pembuatan program, berikut beberapa istilah yang digunakan untuk menyiapkan variabel dan objek aplikasi server.

- Payload : Objek untuk menampung data yang diterima dari *client.*
- PowerPane dll : Objek untuk mengolah chart yang ada di tampilan *dashboard.*
- ListVoltage dll : Merupakan daftar data yang dugunakan untuk mengisi chart, daftar ini yang nanti akan ditambahkan dengan data yang baru.
- Markers : Objek untuk menampung penanda di maps.
- Center : Merupakan penanda untuk lokasi perangkat *wave buoy*.
- Lastmapposchange : Merupakan penanda kapan terakhir kali posisi perangkat berpindah.
- NewPos : Objek yang digunakan untuk menampung koordinat yang diterima dari *client* untuk ditampilan di peta.
- DataCount : Digunakan untuk menghitung jumlah data yang diterima dari *client*.
- Path : Merupakan lokasi untuk menyimpan *chace* peta dan merujuk kepada database aplikasi.

```
using shipmonitoring.Properties;
using System;
using System.Drawing;
using System.IO.Ports;
using System.Windows.Forms;
using shipmonitoring.Data;
using Newtonsoft.Json;
using ZedGraph;
using GMap.NET.WindowsForms;
using GMap.NET.WindowsForms.Markers;
using GMap.NET;

namespace shipmonitoring.Dashboard
{
    public partial class Main : Form
    {

Payload payload;
        GraphPane powerPane, wavePane, tempPane;
        RollingPointPairList listVoltage, listCurrent, listWHeight,
listWPeriod, listWPower, listWater, listAir;
        GMapOverlay markers;
        GMapMarker center;
        DateTime lastmapposchange;
        PointLatLng newPos;

        int dataCount = 0;
        readonly              string              path              =
System.IO.Path.Combine(Environment.GetFolderPath(Environment.Special
Folder.MyDocuments), Application.ProductName);
public Main()
        {
            InitializeComponent();
        }
```

Selanjutnya adalah menyiapkan program untuk menghandle saat form baru dibuka karena program c# memiliki keamanan yang menyebabkan variabel tidak boleh dirubah dari *thread* yang baru. Hal ini akan menyebabkan error jika data dari serial langsung dirubah dalam bentuk chart. Dan perintah "new Payload()" digunakan jika belum ada folder shipmonitoring pada MyDocument yang digunakan untuk penyimpanan database.

```
private void Main_Load(object sender, EventArgs e)
        {

CheckForIllegalCrossThreadCalls = false;

payload = new Payload();
            payload.CreateDir();

SetupPane();
            SetupMap();
            txtStatusKoneksi.Text = Resources.koneksi_siap;
        }
```

4.6.1.2 Program untuk tampilan peta

```
private void SetupMap()
        {

markers = new GMapOverlay("markers");
            newPos = new PointLatLng(-8.240101, 111.469429);
            center          =          new          GMarkerGoogle(newPos,
GMarkerGoogleType.yellow_dot);
            UpdateMapPosition(newPos);

            markers.Markers.Add(center);
            gMapControl1.Overlays.Add(markers);

gMapControl1.CacheLocation = path + "//gmapcache//";
            gMapControl1.MinZoom = 0;
            gMapControl1.MaxZoom = 24;
            gMapControl1.Zoom = 3;

gMapControl1.OnMapZoomChanged += GMapControl1_OnMapZoomChanged;
            gMapControl1.DisableFocusOnMouseEnter = true;

            gMapControl1.RoutesEnabled = false;
            gMapControl1.PolygonsEnabled = false;

            gMapControl1.EmptyTileColor = Color.Gray;
            gMapControl1.HoldInvalidation = true;
gMapControl1.MapProvider                                                    =
GMap.NET.MapProviders.BingMapProvider.Instance;
            GMaps.Instance.Mode = AccessMode.ServerAndCache;
```

Selanjutnya adalah program untuk menghandle ketika posisi peta berubah. Digunakan perintah “invoke” untuk meminta pergantian data dari thread yang baru saja dibuat oleh serial sehingga posisi peta berubah. Dan diperlukan mengecek apakah posisi maps ber[indah terlalu sering atau tidak dengan perintah “if(lastmapposchange.Second !)”.

```
}
private void UpdateMapPosition(PointLatLng currentLoc)
        {

BeginInvoke((Action)delegate
            {
                try
                {

if(lastmapposchange.Second != DateTime.Now.Second)
```

Cek apakah posisi maps berpindah terlalu jauh atau tidak, jika tidak maka posisi penanda tidak perlu berubah.

```
{
if (Math.Abs(currentLoc.Lat - gMapControl1.Position.Lat) > 0.0001 ||
Math.Abs(currentLoc.Lng - gMapControl1.Position.Lng) > 0.0001){
                            gMapControl1.Position = currentLoc;
                        }
```

Reset waktu terakhir ketika posisi maps berubah.

```
  }
lastmapposchange = DateTime.Now;
                    }
```

Merubah posisi penanda ketika posisi map berubah.

```
  }
private void GMapControl1_OnPositionChanged(PointLatLng point)
        {
            center.Position = point;
        }

private void GMapControl1_OnMapZoomChanged()
        {
```

4.6.1.3 Menyiapkan tampilan grafik untuk aplikasi

Proses menyiapkan tampilan grafik untuk aplikasi dimulai dari men-*setup* grafik agar siap untuk diberi data baru; mengatur object powerPane sesuai dengan tampilan dashboard; mengubah text untuk judul, axis x, dan axis y; inisialisasi daftar data tegangan dan arus; memberi nama dan warna pada list data dan memberi tanda kepada grafik bahwa data ada yang dirubah.

```
 private void SetupPane()
        {
powerPane = zedGraphPower.GraphPane;
            powerPane.Title.Text = "Power Consumtion";
            powerPane.XAxis.Title.Text = "Time";
            powerPane.YAxis.Title.Text = "Ampere/Volt";

listVoltage = new RollingPointPairList(1200);
            listCurrent = new RollingPointPairList(1200);

powerPane.AddCurve("Voltage",         listVoltage,         Color.Red,
SymbolType.None);
            powerPane.AddCurve("Current",  listCurrent,  Color.Blue,
SymbolType.None);

zedGraphPower.AxisChange();

            wavePane = zedGraphWave.GraphPane;
            wavePane.Title.Text = "Wave Monitoring";
            wavePane.XAxis.Title.Text = "Time";
            wavePane.YAxis.Title.Text = "Wave";

            listWHeight = new RollingPointPairList(1200);
            listWPeriod = new RollingPointPairList(1200);
            listWPower = new RollingPointPairList(1200);
            wavePane.AddCurve("Wave Height", listWHeight, Color.Red,
SymbolType.None);
            wavePane.AddCurve("Wave       Period",       listWPeriod,
Color.Blue, SymbolType.None);
            wavePane.AddCurve("Wave        Power",        listWPower,
Color.Yellow, SymbolType.None);
zedGraphWave.AxisChange();
```

```
            tempPane = zedGraphTemp.GraphPane;
            tempPane.Title.Text = "Temp Monitoring";
            tempPane.XAxis.Title.Text = "Time";
            tempPane.YAxis.Title.Text = "Celcius";

            listWater = new RollingPointPairList(1200);
            listAir = new RollingPointPairList(1200);
            tempPane.AddCurve("Water  Temp",  listWater,  Color.Red,
SymbolType.None);
            tempPane.AddCurve("Air   Temp",   listAir,   Color.Blue,
SymbolType.None);

            zedGraphTemp.AxisChange();
        }
```

4.6.1.4 Mengatur combo box untuk port yang tersedia

Saat ingin menampilkan sebuah data dari *client* maka diperluka *server* yang dihubungkan ke perangkat computer. Perangkat computer perlu mengenali dimana perangkat *server* diletakkan. Sehingga diperlukan program untuk mengatur combo box saat aplikasi akan dijalankan.

```
private void CmbPort_Click(object sender, EventArgs e)
        {
  UpdatePort();
        }

void UpdatePort()
        {
            cmbPort.Items.Clear();
            cmbPort.Text = String.Empty;
            try
            {
                if (serialPort.IsOpen) serialPort.Close();
            }
            catch { }

            foreach (var port in SerialPort.GetPortNames())
            {
                cmbPort.Items.Add(port);
            }
        }
```

4.6.1.5 Mengatur button koneksi

Ketika port atau lokasi perangkat *server* sudah diketahui saatnya menghubungkan perangkat *server* dengan aplikasi dashboard. Sehingga perlu tombol/*button* untuk menghubungkan.

```
private void BtnKoneksi_Click(object sender, EventArgs e)
        {

if(btnKoneksi.Text.Equals(Resources.button_koneksi_terputus))
            {
                if (cmbPort.Text.Equals(string.Empty))
                {

MessageBox.Show(Resources.warning_comport_kosong);
                    return;
                }

                serialPort.PortName =
cmbPort.SelectedItem.ToString();

                try
                {
                    serialPort.Open();
                    btnKoneksi.Text =
Resources.button_koneksi_tersambung;
                    txtStatusKoneksi.Text =
Resources.koneksi_tersambung;
                    MessageBox.Show(Resources.koneksi_tersambung);
                    cmbPort.Enabled = false;
                    return;
                }
                catch (Exception ex)
                {
                    MessageBox.Show(Resources.warning_koneksi_gagal
+ " (" + ex.Message + ")");
                    Console.WriteLine(ex.Message);
                }
            }
```

```
if (btnKoneksi.Text.Equals(Resources.button_koneksi_tersambung))
            {
                try
                {
                    if (serialPort.IsOpen) serialPort.Close();
                    btnKoneksi.Text =
Resources.button_koneksi_terputus;
                    txtStatusKoneksi.Text =
Resources.koneksi_terputus;
                    cmbPort.Enabled = true;
                }
                catch
                { }
            }
        }
```

4.6.1.6 Mengatur data yang masuk

Setelah perangkat *server* terkoneksi dengan aplikasi maka perlu diatur data yang masuk harus menempati grafik bagian mana.

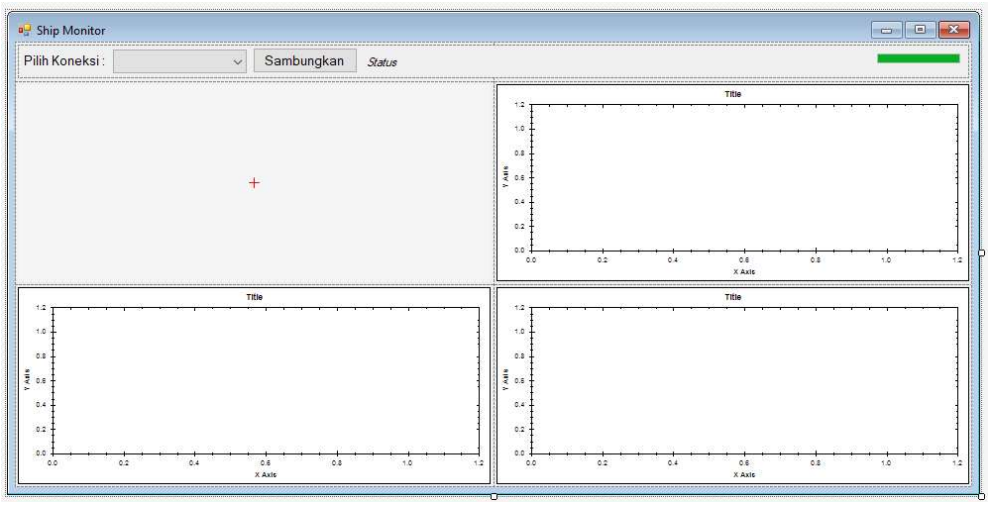
```
private void SerialPort_DataReceived(object sender,
SerialDataReceivedEventArgs e)
        {
          String data = serialPort.ReadLine();
          try
        {
        payload = JsonConvert.DeserializeObject<Payload>(data);
        payload.SaveToDB();
        dataCount++;
        listVoltage.Add(dataCount, payload.Voltage);
        listCurrent.Add(dataCount, payload.Current);

        listWHeight.Add(dataCount, payload.WaveHeight);
        listWPeriod.Add(dataCount, payload.WavePeriod);
        listWPower.Add(dataCount, payload.WavePower);


        listWater.Add(dataCount, payload.WaterTemp);
        listAir.Add(dataCount, payload.AirTemp);
```

Lalu dibuat perintah ketika ada data baru harus selalu update dan diberikan indikasi jika datanya benar maka indikator lampu hijau akan menyala dan jika salah maka indikator lampu merah menyala.

```
        zedGraphPower.AxisChange();
        zedGraphPower.Invalidate();

        zedGraphWave.AxisChange();
        zedGraphWave.Invalidate();

        zedGraphTemp.AxisChange();
        zedGraphTemp.Invalidate();

        newPos.Lat = payload.Latitude;
        newPos.Lng = payload.Longitude;

        UpdateMapPosition(newPos);
        progressBar1.ForeColor = Color.Green;
        }
```

4.6.1.7 Tampilan Dashboard

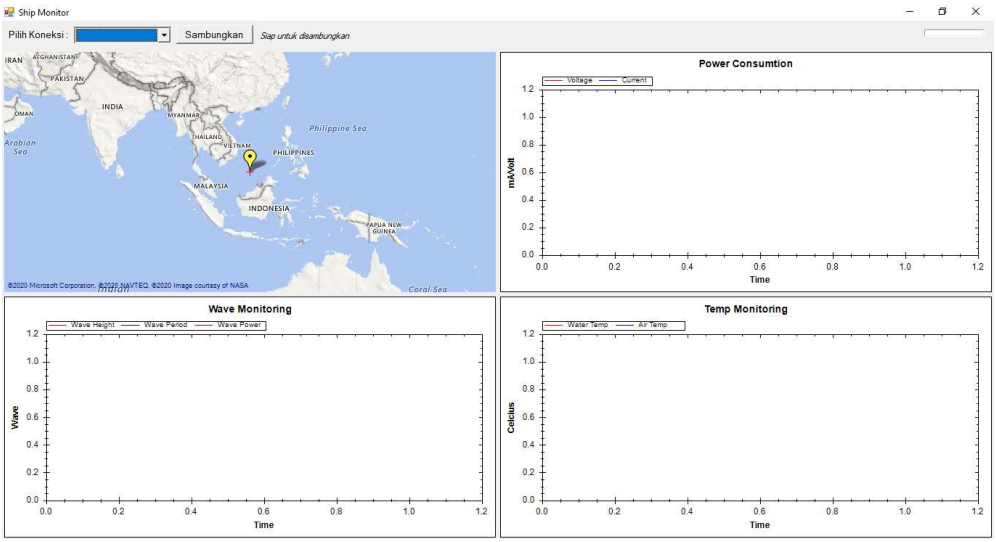
Setelah melalu proses desain maka tampilan dashboard untuk aplikasi *server* adalah seperti yang terlihat pada Gambar 4.8 dan 4.9



Gambar 4.9 Tampilan awal dashboard

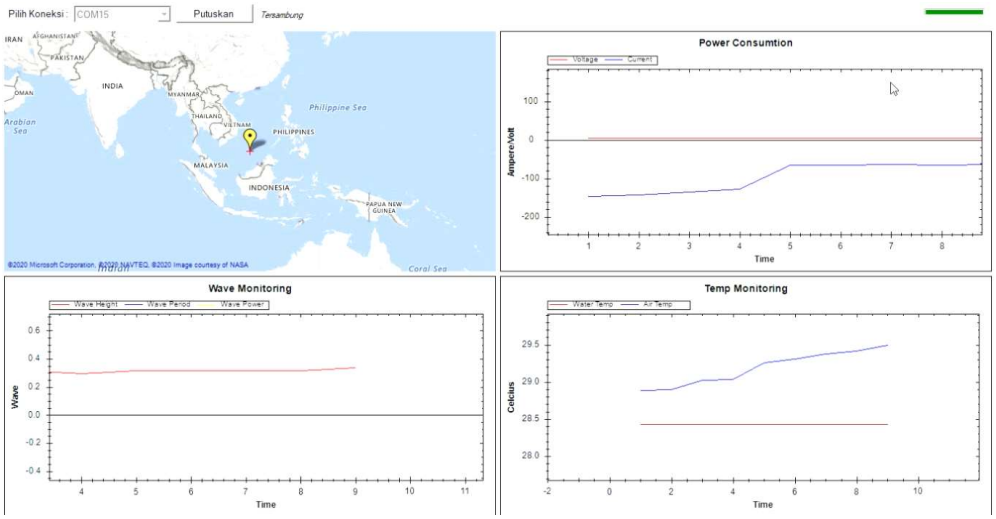
Pada Gambar 4.8 merupakan tampilan awal dashboard sebelum dilakukan running sehingga hanya terlihat tampilan peletakan grafik tetapi belum terdapat identitas setiap grafik.

Sedangkan pada Gambar 4.9 adalah tampilan dashboard setelah dilakukan *running* tetapi belum terhubung dengan perangkat *server.* Terlihat bahwa sudah terdapat keterangan pada setiap grafik dan gambar peta sudah muncul.



Gambar 4.10 Tampilan dashboard setelah running

Dan ketika disambungkan ke perangkat *server* tampilannya akan seperti Gambar 4.10.



Gambar 4.11 Tampilan dashboard ketika *client* dalam keadaan diam dan proses *charging*.

Pada Gambar 4.10 merupakan kondisi dimana alat ukur dalam keadaan diam dan dalam proses pengisian daya. Sehingga grafik “Wave Monitoring” tidak menunjukkan perubahan yang signifikan dalam data ketinggian gelombang dan pada grafik”Power Monitoring” menunjukkan perubahan arus yang semula arus

menunjukkan angka dibawah -100 mA dan mengalami pergerakan naik seiring semakin terisinya daya dari baterai.

4.6.2 Membuat Program Penyimpanan Database

Program penyimpanan database atau payload.cs merupakan sebuah program yang digunakan untuk menampung data yang dikirimkan oleh *client* seperti data waktu, tanggal, gps, gelombang, temperature air, arus dan tegangan baterai. Data ini nanti akan disimpan ke dalam file dengan format Microsoft Acces.

4.6.2.1 Menentukan urutan variabel data

Pertama adalah dengan mendeklarasikan variable data sesuai dengan urutan pada *client.* Dan perintah "get; set;" artinya adalah bahwa data yang diterima dan diolah merupakan data dari luar *(client).*

```
using shipmonitoring.Properties;
using System;
using System.IO;
using System.Windows.Forms;
using System.Data.OleDb;
using Newtonsoft.Json;

namespace shipmonitoring.Data
{
    class Payload
    {
        private String Date { get; set;  }
        private String Time { get; set; }
        public float Latitude { get; set; }
        public float Longitude { get; set; }
        public float Current { get; set; }
        public float Voltage { get; set; }
        public float WaveHeight { get; set; }
        public float WavePeriod { get; set; }
        public float WavePower { get; set; }
        public float WaterTemp { get; set; }
        public float AirTemp { get; set; }
```

4.6.2.2 Menentukan lokasi penyimpanan database

Selanjutnya adalah menentukan lokasi penyimpanan data yaitu pada MyDocument/shipmonitoring.

```
        readonly              string              path              =
System.IO.Path.Combine(Environment.GetFolderPath(Environment.SpecialFold
er.MyDocuments), Application.ProductName);
        OleDbConnection conn;

public void CreateDir()
        {
            if                                  (!Directory.Exists(path))
System.IO.Directory.CreateDirectory(path);
        }

public bool ReadFromDB()
        {
            if (ConnectDB())
            {
                try
                {
                    OleDbCommand cmd = conn.CreateCommand();
                    cmd.CommandText   =   "Select   *   from   Data   where
Date(Tanggal) >= Date(now())";
                    using (OleDbDataReader reader = cmd.ExecuteReader())
                    {
                        while (reader.Read())
                        {
                            string pld = "";
                            JsonConvert.SerializeObject(pld);
                        }
                        reader.Close();
                    }
                    conn.Close();
                    return true;
                }
                catch (Exception)
                {

Console.WriteLine(Resources.warning_database_baca_gagal);
                    return false;
                }
            }
            return false;
        }
```

4.6.3 Mengatur dan menyimpan database

Selanjunya adalah program untuk mengembalikan nilai Boolean.

```
public bool SaveToDB()
        {
            this.Date = DateTime.Now.ToString("dd-MM-yyyy");
            this.Time = DateTime.Now.ToString("hh:mm:ss tt");
```

Penggunaan “if” artinya menjalankan perintah koneksi ke database, jika berhasil maka dilanjutkan ke proses penyimpanan. Dan digunakan “try” agar ketika terjadi error maka aplikasi tidak langsung keluar melainkan akan muncul peringatan.

```
    if (ConnectDB())
            {

try
                {
```

Selanjutnya adalah mengatur pengisian data ke database. Objek “cmd” digunakan untuk menjalankan sql. Perintah “String.Format()” digunakan untuk mengatur urutan data yang akan disimpan ke database. Lalu diberikan perintah “cmd.ExecuteNonQuery()” untuk mengeksekusi perintah sql. Dan yang terakhir ditutup dengan perintah “conn.Close()”.

```
OleDbCommand cmd = conn.CreateCommand();

cmd.CommandText = String.Format("Insert into tbl_data(Tanggal, Waktu,
Latitude, Longitude, [Current], Voltage, WaveHeight, WavePeriod,
WavePower, WaterTemp, AirTemp) values('{0}', '{1}', '{2}', '{3}',
'{4}', '{5}', '{6}', '{7}', '{8}', '{9}', '{10}')",
                        this.Date,
                        this.Time,
                        this.Latitude,
                        this.Longitude,
                        this.Current,
                        this.Voltage,
                        this.WaveHeight,
                        this.WavePeriod,
                        this.WavePower,
                        this.WaterTemp,
                        this.AirTemp);

cmd.ExecuteNonQuery();

conn.Close();
```

Setelah berhasil disimpan berikan tanda bahwa data sudah berhasil disimpan.

```
Console.WriteLine(Resources.warning_database_simpan_berhasil);
                    return true;
                }
                catch (Exception ex)
                {

Console.WriteLine(Resources.warning_database_simpan_gagal     +     "
,Message: " + ex.Message);
                    return false;
                }
            }
            return false;
        }
```

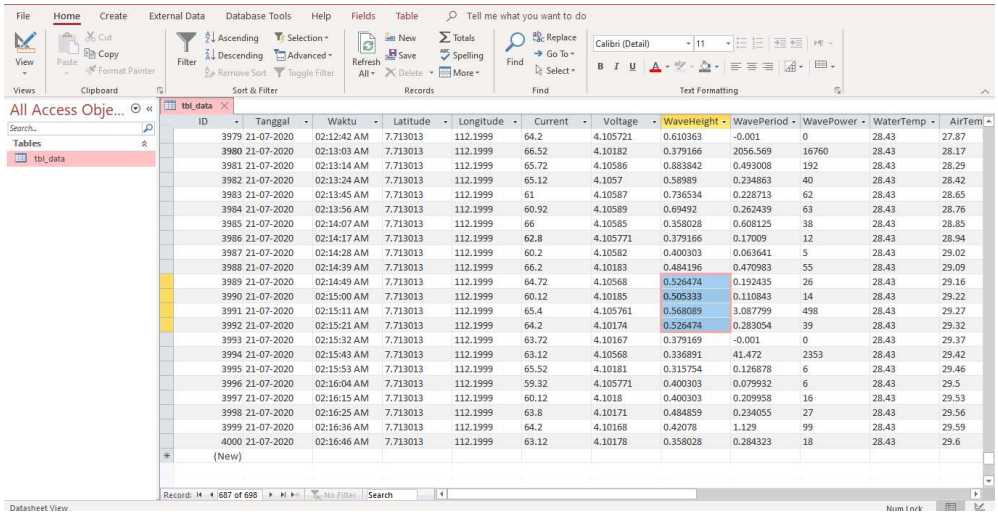
Langkah terakhir berikan perintah kembali ke database.

```
private bool ConnectDB()
        {

  string connectionString = @"Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=" + path + "\\Database.accdb";
            conn = new OleDbConnection(connectionString);
try
            {
                conn.Open();
                return true;
            }
            catch (Exception ex)
            {

MessageBox.Show(Resources.warning_database_koneksi_gagal);
                Console.WriteLine(ex.Message);
                return false;
            }
        }
    }
}
```

4.7 Pengujian Alat Ukur

Pengujian alat ukur menggunakan alat simulator sederhana yang digunakan untuk menguji sensor ms5611 dalam membaca perubahan ketinggian. Cara kerja alat simulator mirip dengan cara kerja jungkat-jungkit, dimana alat ukur nantinya akan diletakkan di salah satu sisi lalu sisi lainnya memberikan gerakan naik turun.

Alat simulator memiliki tinggi maksimal 110 cm yang diukur dari dasar lantai hingga titik terjauh alat tersebut dapat mengangkat alat ukur. Sehingga jika proses pengukuran presisi akan menghasilkan ketinggian gelombang 55 cm atau 0.55 m. Hasil pengujian dalam bentuk database pada McAcces dapat dilihat pada Gambar 4.11.



ID	Tanggal	Waktu	Latitude	Longitude	Current	Voltage	WaveHeight	WavePeriod	WavePower	WaterTemp	AirTem
3979	21-07-2020	02:12:42 AM	7.713013	112.1999	64.2	4.105721	0.610363	-0.001	0	28.43	27.87
3980	21-07-2020	02:13:03 AM	7.713013	112.1999	66.52	4.10182	0.379166	2056.569	16760	28.43	28.17
3981	21-07-2020	02:13:14 AM	7.713013	112.1999	65.72	4.10586	0.883842	0.493008	192	28.43	28.29
3982	21-07-2020	02:13:24 AM	7.713013	112.1999	65.12	4.1057	0.58989	0.234863	40	28.43	28.42
3983	21-07-2020	02:13:45 AM	7.713013	112.1999	61	4.10587	0.736534	0.228713	62	28.43	28.65
3984	21-07-2020	02:13:56 AM	7.713013	112.1999	60.92	4.10589	0.69492	0.262439	63	28.43	28.76
3985	21-07-2020	02:14:07 AM	7.713013	112.1999	66	4.10585	0.358028	0.608125	38	28.43	28.85
3986	21-07-2020	02:14:17 AM	7.713013	112.1999	62.8	4.105771	0.379166	0.17009	12	28.43	28.94
3987	21-07-2020	02:14:28 AM	7.713013	112.1999	60.2	4.10582	0.400303	0.063641	5	28.43	29.02
3988	21-07-2020	02:14:39 AM	7.713013	112.1999	66.2	4.10183	0.484196	0.470983	55	28.43	29.09
3989	21-07-2020	02:14:49 AM	7.713013	112.1999	64.72	4.10568	0.526474	0.192435	26	28.43	29.16
3990	21-07-2020	02:15:00 AM	7.713013	112.1999	60.12	4.10185	0.505333	0.110843	14	28.43	29.22
3991	21-07-2020	02:15:11 AM	7.713013	112.1999	65.4	4.105761	0.568089	3.087799	498	28.43	29.27
3992	21-07-2020	02:15:21 AM	7.713013	112.1999	64.2	4.10174	0.526474	0.283054	39	28.43	29.32
3993	21-07-2020	02:15:32 AM	7.713013	112.1999	63.72	4.10167	0.379169	-0.001	0	28.43	29.37
3994	21-07-2020	02:15:43 AM	7.713013	112.1999	63.12	4.10568	0.336891	41.472	2353	28.43	29.42
3995	21-07-2020	02:15:53 AM	7.713013	112.1999	65.52	4.10181	0.315754	0.126878	6	28.43	29.46
3996	21-07-2020	02:16:04 AM	7.713013	112.1999	59.32	4.105771	0.400303	0.079932	6	28.43	29.5
3997	21-07-2020	02:16:15 AM	7.713013	112.1999	60.12	4.1018	0.400303	0.209958	16	28.43	29.53
3998	21-07-2020	02:16:25 AM	7.713013	112.1999	63.8	4.10171	0.484859	0.234055	27	28.43	29.56
3999	21-07-2020	02:16:36 AM	7.713013	112.1999	64.2	4.10168	0.42078	1.129	99	28.43	29.59
4000	21-07-2020	02:16:46 AM	7.713013	112.1999	63.12	4.10178	0.358028	0.284323	18	28.43	29.6
	(New)										

Gambar 4.12 Hasil pengujian alat ukur dalam bentuk database Mc.Acces

4.8 Analisa Hasil Pengujian

4.8.1 Analisa Hasil Pengujian Data Gelombang

Dalam menganalisa hasil pengujian data gelombang diambil acuan pada pengukuran ketinggian gelombang dengan menggunakan data yang mendekati nilai maksimal 0,55 m sesuai spesifikasi simulator. Sehingga diambil data pada ID 3989 hingga 3992 sesuai Gambar 4.11 dan disajikan dalam Tabel 4.2.

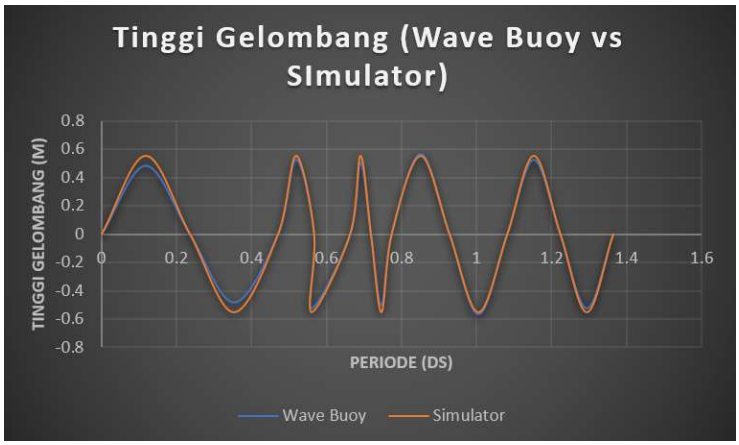
Tabel 4.2 Data Pengujian Ketinggian Gelombang

ID Data	*Wave Height* (m)
3989	0.526474
3990	0.505333
3991	0.568089
3992	0.526474

Dalam Gambar terlihat bahwa pada ID data 3989 hingga 3992 menunjukkan nilai yang mendekati nilai maksimal yang dapat diukur dari pengujian menggunakan simulator. Sehingga dapat dihitung error dalam pengukuran ketinggian gelombang dengan menggunakan perhitugan sebagai berikut:

$Error_{WH}$ = (0.55 - (Σ($Data_{3989-3992}$)/4) / 0.55) x 100%
= (0.55-0.53)/0.55 x 100%
= 3.6%

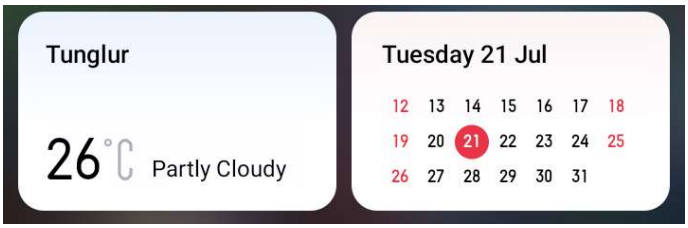
Dan jika ditampilkan dalam sebuah grafik maka akan diperoleh tampilan sebagai berikut:



Gambar 4.13 Grafik Tinggi Gelombang (Wave Buoy vs Simulator)

4.8.2 Analisa Hasil Pengujian Temperatur Udara

Untuk menganalisa hasil pengujian temperature udara digunakan acuan data temperatur yang dihasilkan dari pengukuran melalui *smartphone* yang menunjukkan angka 26ºC sesuai Gambar 4.12.



Gambar 4.14 Pengukuran temperatur udara melalui *smartphone.*

Dan hasil pengukuran temperatur udara selama pengujian dapat dilihat pada Tabel 4.3.

Tabel 4.3 Hasil pegukuran temperatur udara selama pengujian.

No	ID Data	AirTemp
1	3980	28.17
2	3981	28.29
3	3982	28.42
4	3983	28.65
5	3984	28.76
6	3985	28.85
7	3986	28.94
8	3987	29.02
9	3988	29.09
10	3989	29.16
11	3990	29.22
12	3991	29.27
13	3992	29.32
14	3993	29.37
15	3994	29.42
16	3995	29.46
17	3996	29.5
18	3997	29.53
19	3998	29.56
20	3999	29.59

21	4000	29.6
Rata-Rata		**29.1043**

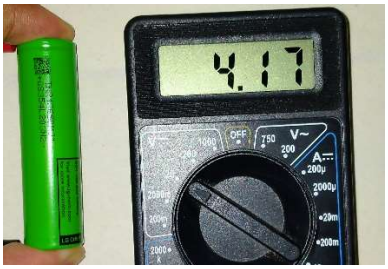
Dengan demikian dapat dihitung error dalam pengukuran temperature udara dengan menggunakan perhitungan sebagai berikut:

$Error_{AT}$ = |(26-29.1)/26| X 100%

= 11.9%

4.8.3 Analisa Hasil Pengujian Voltase Baterai

Analisa hasil pengujian voltase baterai yaitu dengan mencocokan hasil tegangan baterai yang diukur menggunakan avometer digital dengan hasil pengukuran selama pengujian. Hasil pengukuran baterai dengan menggunakan avometer dapat dilihat pada Gambar 4.13.



Gambar 4.15 Pengukuran tegangan baterai menggunakan avometer digital

Dan nilai tegangan baterai rata-rata selama proses pengujian dapat dilihat dalam Tabel 4.4.

Tabel 4.4 Hasil pengukuran tegangan baterai

No	ID Data	Voltage
1	3980	4.10182
2	3981	4.10586
3	3982	4.1057
4	3983	4.10587
5	3984	4.10589
6	3985	4.10585
7	3986	4.105771
8	3987	4.10582
9	3988	4.10183
10	3989	4.10568
11	3990	4.10185

12	3991	4.105761
13	3992	4.10174
14	3993	4.10167
15	3994	4.10568
16	3995	4.10181
17	3996	4.105771
18	3997	4.1018
19	3998	4.10171
20	3999	4.10168
21	4000	4.10178
Rata-Rata		**4.10387**

Sehingga error dalam pengukuran tegangan baterai selama pengujian dapat dihitung dengan menggunakan perhitungan sebagai berikut:

$Error_{BV}$ = (4.17-4.103)/4.17 x 100%

= 1.59%

4.9 Biaya Pembuatan

Rincian biaya pembuatan pembangunan sistem elektronik alat ukur ketinggian air laut dengan menggunakan modul sensor GY 86 dapat dilihat dalam Tabel 4.5.

Tabel 4.5 Rincian Biaya

Komponen	Jumlah	Harga satuan	Harga x Jumlah	Referensi
Arduino Nano	2	45.000	90.000	Tokopedia/akhi_shop
GY 86	1	310.000	310.000	Tokopedia/Beetrona
TCA9548A	1	65.400	65.400	Tokopedia/AIF Robotik
INA219	1	40.000	40.000	Tokopedia/ISee
GPS Neo6MV2	1	85.000	85.000	Tokopedia/akhi_shop
DS18B20	1	20.000	20.000	Tokopedia/akhi_shop
NRF24L01	2	40.000	80.000	Tokopedia/akhi_shop
XL6009	1	25.000	25.000	Tokopedia/digpart
TP4056	1	6000	6000	Tokopedia/YoyokArjom
LG INR18650MJ1	1	70.000	70.000	Tokopedia/tokojelek#1
Battery Holder	1	25.000	25.000	Toko Jago
PCB	1	204.800	204.800	Toko Maztron
Total Biaya			**1.021.200**	

BAB 5 PENUTUP

5.1 Kesimpulan

A. Desain alat ukur

Desain alat ukur ketinggian gelombang air laut dengan menggunakan modul sensor GY86 secara umum dibagi menjadi sistem penginderaan tekanan dengan menggunakan sensor MS5611 yang merupakan sensor barometer; sistem pengiriman data menggunakan modul radio NRF24L01 dan disimpan pada sebuah database McAcces dengan interface yang menggunakan software Visual Studio; dan sistem monitoring tegangan baterai menggunakan sensor arus INA219.

Alat ukur ketinggian gelombang menggunakan metode pengukuran ketinggian dengan acuan perubahan tekanan udara relatif dengan menggunakan sensor MS5611 yang terdapat pada modul sensor GY86 menghasilkan error sebesar 3.6%. Sedangkan pengukuran temperatur udara dengan menggunakan sensor MS5611 menghasilkan error 11.6% dan pengukuran tegangan baterai menggunakan sensor INA219 menghasilkan error 1.6%.

B. Biaya pembuatan alat ukur

Dari keseluruhan desain sistem elektronik yang telah dibuat dibutuhkan biaya pembuatan alat sebesar Rp.1.021.200,00. Dengan demikian harga tersebut masih sangat terjangkau bagi peneliti dan institusi.

5.2 Saran

Selama proses pembuatan alat ukur ketinggian air laut dengan menggunakan modul sensor GY 86 terdapat kendala atau masalah yang dapat digunakan sebagai acuan untuk penelitian lebih lanjut mengenai alat ukur ketinggian air laut.

Pertama merupakan metode pengukuran yang berbasis tekanan menggunakan sensor ms5611 masih memiliki error yang cukup besar dikarenakan acuan dari perubahan ketinggian adalah tekanan atmosfir relative yang sangat sensitive sehingga kurang presisi. Sehingga untuk kedepan mungkin bisa menggunakan sensor yang lebih baik selain menggunakan barometer.

Kedua dalam segi desain sistem awalnya didesain menggunakan modul SD Card untuk sistem *backup* penyimpanan data ketika radio komunikasi berada diluar jangkauan sehingga perangkat *server* tidak bisa menerima data. Tetapi hal ini mengalami kegagalan dikarenakan modul SD Card menggunakan SPI sama dengan NRF24L01 sehingga ketika modul SD Card digunakan maka NRF24L01 akan mengalami kegagalan dalam proses *initializing* perangkat. Oleh sebab itu untuk penelitian selanjutnya bisa dipertimbangkan sebuah sistem yang bisa mengirim sekaligus mem-*backup* data melalui SD Card.

"*Halaman Ini Sengaja Dikosongkan"*

DAFTAR PUSTAKA

[1] "eiva," EIVA, 27 Mei 2020. [Online]. Available: www.eiva.com/products/hardware/buoys-and-floats/toughboy-wave-buoy/toughboy-panchax-1-2-with-1200-khz-adcp. [Diakses 18 Juli 2020].

[2] E. Munandar, I. Jaya dan A. S. Atmadipoera, "Rancang Bangun Uji Kinerja Wave Buoy sebagai Alat Pengukur Tinggi Gelombang Pesisir," *Jurnal Ilmu dan Teknologi Kelautan Tropis,* vol. 10, pp. 1-14, 2018.

[3] B. Triatdmojo, Teknik Pantai, Yogyakarta: Beta Offshet, 1999.

[4] A. Supangat dan Susanna, Pengantar oseanografi, Jakarta: Badan Riset Kelautan dan Perikanan, 2003.

[5] K. Wyrkti, "Phyical Oceanography of the South East Asian Waters," Naga Report, California, 1961.

[6] J. I. Pariwono, dalam *Kondisi pasang-surut di Indonesia*, Jakarta, Pusat Penelitian dan Pengembangan Oseanograf, 1987, pp. 135-147.

[7] D. T. Pugh, Tides, surges, and mean sea level, Wiltshire: John Wiley & Sons Ltd, 1987.

[8] O. Philip, Dinamika Laut Atas (edisi II), Cambridge University Press, 1977.

[9] M. Tucker, Gelombang dalam rekayasa kelautan, Oxford Elseveir, 2001.

[10] "How to measure absolute pressure using piezoresistif sensing elements," AMSYS GmbH & Co. KG, Mainz Germany.

[11] "MS5611 - Precise altitude measurement with pressure sensors," AMSYS GmbH & Co.KG, Mainz Germany.

[12] U. S. Utara, "repository.usu.ac.id," 2018. [Online]. Available: http://repository.usu.ac.id/bitstream/handle/123456789/62649/Chapter%20II.pdf?sequence=4&isAllowed=y. [Diakses 27 Januari 2020].

[13] D. Kho, "teknikelektronika," 2019. [Online]. Available: https://teknikelektronika.com/pengertian-multiplexer-multiplekser-cara-kerja-multiplexer/. [Diakses 27 Januari 2020].

[14] Splashtronic, “splashtronic,” 29 Agustus 2013. [Online]. Available: https://splashtronic.wordpress.com/2013/10/29/modul-sd-card/. [Diakses 27 Januari 2020].

[15] Wikipedia, “Wikipedia,” [Online]. Available: https://id.wikipedia.org/wiki/Kartu_Secure_Digital.

[16] P. N. Sriwijaya, “http://eprints.polsri.ac.id/,” [Online]. Available: http://eprints.polsri.ac.id/2861/3/FILE%20III%20%28BAB2%29.pdf.

[17] D. Tekno, “De Tekno,” [Online]. Available: https://de-tekno.com/2018/05/mengenal-battery-18650-bettery-dengan-power-besar/.

[18] O. S. Ongkosongo, “Penerapan pengetahuan data pasang-surut,” Jakarta, LIPI Pusat Penelitian dan Pengembangan Oseanografi, pp. 241-255.

[19] D. Bonvik dan Stone, “miros-group.com,” 4 Februari 2019. [Online]. Available: www.miros-group.com. [Diakses 4 Mei 2020].

[20] “Wikipedia,” 2018. [Online]. Available: www.wikipedia.org. [Diakses 4 Mei 2020].

PROFIL PENULIS

Dennis Chaniago Ramadhan, lahir di Kota Kediri, Jawa Timur pada tanggal 3 Januari 1998. Penulis merupakan anak pertama dari tiga bersaudara. Penulis menempuh pendidikan pada tingkat dasar di TK Dharma Wanita Tunglur Kediri, kemudian melanjutkan pendidikan sekolah dasar di SDN 1 Tunglur Kediri, MTsN 1 Pare Kediri, dan pendidikan menengah atas di MAN 3 Kota Kediri (Sekarang MAN 2 Kota Kediri). Setelah lulus dari pendidikan menengah atas, penulis diterima dan melanjutkan pendidikan perguruan tinggi di Departement Teknik Sistem Perkapalan, Fakultas Teknologi Kelautan, Institut Teknologi Sepuluh Nopember, Surabaya.

Selama menempuh pendidikan di Departemen Teknik Sistem Perkapalan ITS, penulis turut aktif di beberapa kegiatan organisasi kampus. Pada tahun 2017 hingga 2019 penulis aktif dalam sebuah tim riset kapal cepat tenaga surya yang bernama ITS Marine Solar Boat Team sebagai *engineer* pada divisi *electrical engineering*. Selain itu penulis juga aktif dalam kegiatan sosial masyarakat pada tahun 2018 dan 2019 sebagai koordinator *Community Service* yang diadakan oleh Marine Icon dari Himpunan Mahasiswa Teknik Sistem Perkapalan. Pada bidang penelitian, penulis juga pernah mengikuti lomba karya tulis yang diadakan oleh Universitas Andalas Sumatera Barat dengan predikat Juara II dan mendapatkan predikat *Commedation Award* pada sebuah kompetisi karya tulis internasional yang diselenggarakan oleh Sembawang Shipyard, Singapura.

E-mail : dennischaniago763@gmail.com

Kontak : +6287762021576

"*Halaman Ini Sengaja Dikosongkan"*

UCAPAN TERIMAKASIH

Dalam menempuh dan menyelesaikan pendidikan jenjang sarjana pada Departemen Teknik Sistem Perkapalan FTK ITS, penulis mengucapkan terimakasih kepada:

1. Dinda Aurelia Putri selaku partner terdekat yang selalu memberikan dukungan kepada penulis.

2. Muhammad Azmi Naufal Munif dan Muhammad Rifki Abdillah selaku partner penulis dalam bidang kelistrikan.

3. Taruno Ganggas, Prafasta Aru, Muhammad Hadrian, dan Muhammad Fathin selaku partner penulis dalam bidang permesinan.

4. Tatyana A. Ibrahim selaku teman diskusi penulis.

5. Viorel Herniza dan Allysha Prista selaku teman penulis dalam berwisata kuliner di Surabaya.

6. Sahal Adzkiya, Ario Fajar, Ahmad Ulul Albab, Rijal Akmal Syah, Muhammad Naufal, dan Fahmi Syaifuddin selaku teman pondok penulis.

7. Rifqi Rizqullah, Dwicky Rivaldo, dan Fikran Ladamay selaku teman penulis ketika bermalam di kampus.

"*Halaman Ini Sengaja Dikosongkan"*