

BAB III

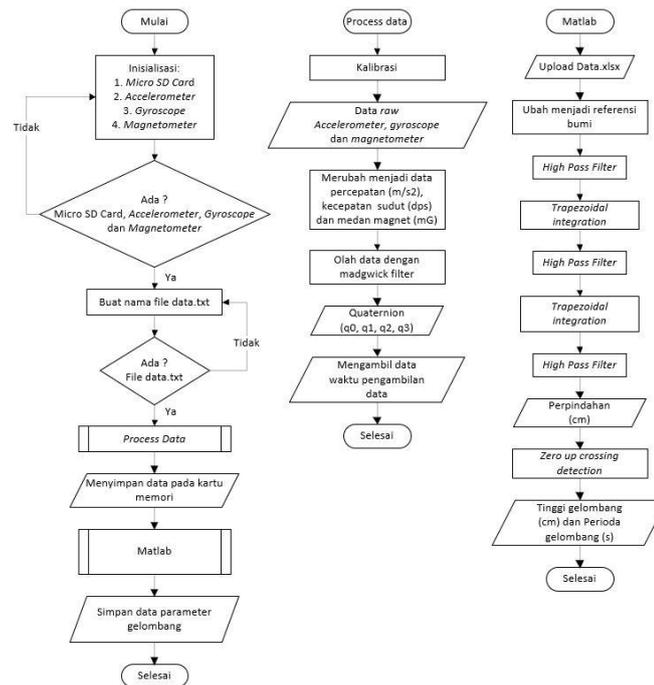
METODE PENELITIAN

3.1 Metode Penelitian

Penelitian dalam skripsi ini menggunakan serangkaian kegiatan yang dilakukan untuk memperoleh tujuan yang ingin dicapai. Kegiatan-kegiatan tersebut harus tersusun dalam sebuah metodologi. Metodologi penelitian ini adalah sebagai berikut:

1. Studi literatur, mencari jurnal, artikel penelitian dan buku-buku yang berhubungan dengan permasalahan yang dibahas pada penelitian yang dapat membantu dalam pelaksanaan penelitian.
2. Merancang *Hardware*, merancang sistem elektrikal MPU9250 dan *micro sd card module* untuk terintegrasi dengan Arduino.
3. Merancang *software*, perancangan *software* Arduino IDE untuk membaca sensor MPU9250, menyimpan data dengan perantara *micro sd card module* juga perancangan *software* matlab untuk analisis data menjadi gelombang.
4. Menguji Sistem, dilakukan untuk mengetahui kinerja sistem yang telah dibuat. Pengujian sistem dilakukan secara simulasi. Pengujian antara lain simulasi statis dan simulasi dinamis, pada simulasi statis dilakukan pengujian sensor MPU9250 dalam kondisi diam sedangkan pengujian simulasi dinamis dilakukan pada saat kondisi bergerak
5. Menganalisis penelitian dan pengujian, berisi analisis hasil dan pembahasan dari sistem yang telah dibuat dan membuat kesimpulan dari pengujian tersebut.

Sensor *accelerometer*, *gyroscope*, dan *magnetometer* yang terdapat pada modul MPU9250 terintegrasi dengan Arduino mega 2560. Data dari ketiga sensor diproses oleh Atmega 2560 sehingga menghasilkan data percepatan, kecepatan sudut dan besar medan magnet. Data disimpan pada kartu memori melalui modul *SD card reader* yang datanya diproses pada laptop menggunakan *software* matlab. Laptop berfungsi sebagai pengolah data dan juga sebagai media penampil informasi karakteristik gelombang laut, kartu memori berfungsi sebagai penyimpanan data. Flowchart penelitian bisa dilihat pada (Gambar 3.1).



Gambar 3. 1 Diagram Alir Sistem

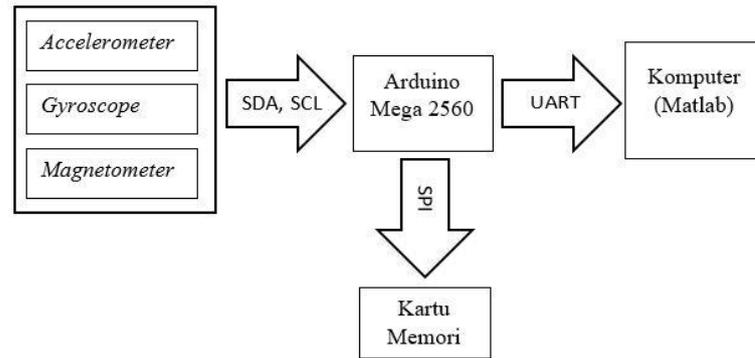
Alur program secara keseluruhan dari sistem pada penelitian ini digambarkan dengan *flowchart* pada (Gambar 3.1). Dimulai dari inisialisasi sensor dan modul. Pengambilan data beserta kalibrasi sensor. Menyimpan data pada kartu memori. Proses filter dan sekaligus menghitung nilai quaternion menggunakan madgwick filter. Hingga mengubah data dari percepatan menjadi perpindahan secara vertikal.

3.2 Instrumentasi Penelitian

Instrument penelitian yang digunakan pada penelitian ini meliputi perangkat keras (*hardware*) yaitu Arduino Mega 2560, MPU9250, *micro sd reader*, kartu memori 8Gb, dan laptop. Perangkat lunak (*software*) yang digunakan adalah Arduino IDE dan matlab.

3.2.1 Perangkat Keras

Perancangan perangkat keras bertujuan untuk akusisi data dari sensor IMU dan mengolah data menjadi data karakteristik gelombang laut. Data yang menjadi perhatian pada penelitian ini adalah data perpindahan alat secara vertikal.. Diagram blok perangkat keras AHRS disajikan pada (Gambar 3.2).



Gambar 3. 2 Diagram Blok Sistem Perangkat Keras

Diagram blok dari perangkat keras pada (Gambar 3.2). Perangkat keras terdiri dari tiga sensor inersia (*accelerometer*, *gyroscope* dan *magnetometer*) yang tergabung dalam sensor MPU9250 sebagai masukan dari sistem yang terhubung dengan mikrokontroler menggunakan komunikasi I²C, mikrokontroler arduino mega 2560 sebagai pengolah data, kartu memori sebagai penyimpan data yang terhubung dengan mikrokontroler menggunakan komunikasi SPI dengan bantuan modul *card reader*, dan laptop sebagai pengolah dan penampil data.

1. Arduino Mega 2560

Penelitian ini menggunakan mikrokontroler Arduino mega 2560. Arduino digunakan untuk mengambil data dari sensor MPU9250, mengolah data hingga mendapatkan quaternion. Arduino Mega 2560 dipilih karena memiliki kristal sebesar 16MHz, sehingga mikrokontroler dapat berkomunikasi menggunakan UART (*Universal Asynchronous Receiver- Transmitter*) ke komputer dengan *baud rate* sebesar 115200bps. Konfigurasi pin dari Arduino Mega lihat (Tabel 3.1).

Tabel 3. 1 Konfigurasi pin Arduino Mega 2560

Pin	Konfigurasi	Keterangan
Vcc	5v	Hubungkan dengan <i>pin</i> Vcc MU9250
Vcc	3.3v	Hubungkan dengan <i>pin</i> Vcc <i>micro sd card reader</i>
GND	Ground	Hubungkan dengan <i>pin</i> GND
D20	SDA	Hubungkan dengan <i>pin</i> SDA sensor IMU 9250
D21	SCL	Hubungkan dengan <i>pin</i> SCL sensor IMU 9250
D50	MISO	Hubungkan dengan <i>pin</i> MISO <i>micro sd card reader</i>
D51	MOSI	Hubungkan dengan <i>pin</i> MOSI <i>micro sd card reader</i>
D52	Clk	Hubungkan dengan <i>pin</i> Clk <i>micro sd card reader</i>
D53	SS	Hubungkan dengan <i>pin</i> SS <i>micro sd card reader</i>

Spesifikasi pada (Tabel 3.1) merupakan daftar pin Arduino Mega 2560 yang digunakan. Pin 5v dan 3.3v merupakan pin yang menghasilkan sumber tegangan listrik yang digunakan sebagai sumber tegangan untuk sensor MPU9250 dan modul *micro sd card reader*. Pin SCL dan SDA adalah pin komunikasi I²C dengan MPU9250, pin MISO, MOSI, Clk dan SS adalah pin komunikasi SPI dengan *micro sd card reader*. Sumber daya untuk Arduino sendiri berasal dari *battery 9v*.

2. MPU9250

MPU9250 pada penelitian ini berfungsi untuk mengambil data percepatan, medan magnet dan kecepatan sudut. Sensor ini memiliki fitur 16-bit ADCs (*analog to digital converters*) dan memiliki faktor skala yang dapat diprogram. Pada *accelerometer* faktor skala antara $\pm 2g$, $\pm 4g$, $\pm 8g$ dan $\pm 16g$, pada *gyroscope* faktor skala antara $\pm 250\text{deg/s}$, $\pm 500\text{deg/s}$, $\pm 1000\text{deg/s}$ dan $\pm 2000\text{deg/s}$ dan *magnetometer* memiliki faktor skala $\pm 4800\mu\text{T}$. MPU9250 memiliki dua pilihan komunikasi yaitu I²C pada 400kHz atau SPI pada frekuensi 20MHZ [27].

Tegangan operasi untuk MPU9250 antara 2.5 – 3.6V, sehingga pin vcc terhubung dengan sumber 3.3V dan pin gnd terhubung dengan pin GND. Komunikasi data dengan mikrokontroler menggunakan I²C, sehingga pin SDA terhubung dengan pin D20 dan pin SCL terhubung dengan pin D21 pada Arduino, SDA (*Serial Data*) berfungsi untuk mengirim data antara *master* (mikrokontroler) dan *slave* (MPU9250) sedangkan SCL (*Serial Clock*) berfungsi untuk mengirim sinyal *clock* [16][27]. Faktor skala *accelerometer* yang pilih adalah $\pm 2g$, *gyroscope* adalah $\pm 250\text{deg/s}$ dan untuk *magnetometer* $\pm 4800\mu\text{T}$.

3. Micro Sd card 8GB

Sd card digunakan untuk menyimpan data dari sensor yang telah diproses oleh mikrokontroler menjadi informasi percepatan (m/s^2), kecepatan sudut (rad/s), medan magnet (mG) serta waktu pencatatan untuk pengambilan setiap data. Informasi yang disimpan pada memori ini dimasukkan pada *software* matlab untuk diproses menghasilkan keluaran yang diinginkan.

4. Modul Micro SD Card

Modul *micro SD card* diperlukan sebagai media penghubung antara kartu memori dengan mikrokontroler. Komunikasi dengan kartu memori menggunakan komunikasi SPI melalui pin MOSI (*Master Output Slave Input*) untuk mengirim

data dari *master* (mikrokontroler) ke *slave* (Modul *Micro SD Card*), MISO (*Master Input Slave Output*) fungsinya untuk mengirim data dari *slave* ke *master*, SCLK (*SPI Clock*) adalah pin untuk sinyal *clock* dan pin SCK (*Slave/Chip Select*) digunakan master untuk memilih *slave* yang dikirim data. Beberapa spesifikasi tambahan dari modul *micro SD card* (Tabel 3.2).

Tabel 3. 2 Modul *Micro SD Card*

Spesifikasi	Nilai
<i>Support Card</i>	<i>Micro SD</i>
Tegangan operasi	4.4 – 5.5 V
Arus operasi	80 – 200 mA

Spesifikasi pada (Tabel 3.2) merupakan informasi mengenai spesifikasi elektrik modul *micro SD card* yang digunakan pada penelitian ini. Sumber tegangan dan arus berasal dari mikrokontroler arduino mega 2560, yaitu pin 5v. Tegangan ini diturunkan oleh regulator yang ada pada modul untuk diberikan sebagai masukan *Vcc* pada *micro sd card*, karena *micro sd card* sendiri hanya membutuhkan tegangan operasi sebesar 3.3v.

3.2.2 Perangkat Lunak

Perangkat lunak yang digunakan untuk memprogram Arduino Mega adalah Arduino IDE. Penggunaan Arduino IDE dengan menulis algoritma dan perhitungan menggunakan bahasa pemrograman kusus, *men-debug* serta memprogram. Ketika digunakan dengan perangkat keras yang terkait, kombinasi perangkat keras-perangkat lunak ini memungkinkan untuk menguji perangkat keras saat melihat dan melakukan *debug* pada perangkat lunak.

1. Arduino IDE

Arduino IDE merupakan perangkat lunak gratis yang resmi dikeluarkan oleh arduino. Fungsi dari program ini adalah melakukan *compile* program, pengisian kode program, dan serial monitor. Arduino IDE digunakan karena semua sensor dan modul yang digunakan terintegrasi dengan mikrokontroler arduino. Keunggulan dari Arduino IDE adalah dapat melakukan pengisian program dan serial monitor yang terpadu dalam satu perangkat lunak ini tidak membutuhkan perangkat lunak lain untuk membaca data *serial*.

2. MATLAB

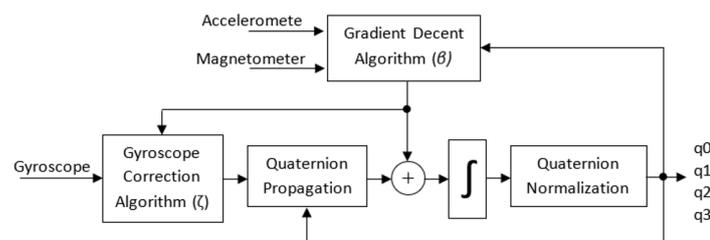
MATLAB merupakan kepanjangan dari *Matrix Laboratory*, perangkat lunak ini memiliki keunggulan untuk menyelesaikan komputasi dalam bidang matematika khususnya untuk melakukan perhitungan yang melibatkan matriks [41]. Penelitian ini menggunakan perangkat lunak MATLAB untuk menjalankan Algoritma *zero crossing* dengan memasukkan data perpindahan vertikal dari arduino, membuat grafik hasil.

3.3 Akuisisi Data Sensor

Akuisisi data sensor MPU 9250 dilakukan dengan menghubungkan antara sensor tersebut ke Arduino Mega 2560. Data yang dikirimkan dari sensor merupakan *raw* data yang terdiri dari 3 sumbu *accelerometer*, 3 sumbu *gyroscope*, dan 3 sumbu *magnetometer*. Terdapat 9 *raw* data yang diambil melalui Arduino Mega. Pengambilan data pertama-tama dilakukan dengan menghubungkan sensor MPU 9250 dengan arduino Mega 2560 yang terkoneksi ke komputer dengan menggunakan *software* IDE untuk menampilkan hasil data pengukuran awal dari *accelerometer*, *gyroscope*, *magnetometer*.

3.4 Perancangan Madgwick Filter

Algoritma filter ini dimulai dengan menormalisasi data dari *accelerometer* dan *magnetometer*, menghitung medan magnet dalam referensi bumi dan efek dari pengukuran kemiringan yang salah terhadap arah medan magnet[41]. Algoritma ini menggabungkan kompensasi distorsi magnetik dan menggunakan representasi quaternion yang memungkinkan data *accelerometer* dan *magnetometer* digunakan dalam algoritma *gradien descent* dan dioptimalkan untuk menghitung kesalahan pengukuran *gyroscope* hasil dari turunan Quaternion (Gambar 3.3).

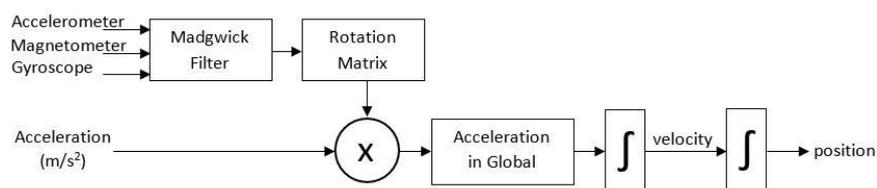


Gambar 3. 3 Diagram Blok Sistem

Algoritma madgwick berdasarkan diagram blok pada (Gambar 3.3) terbagi menjadi dua proses. Pertama, pengukuran *gyroscope* diuraikan dengan algoritma koreksi, yang tergantung pada parameter ζ , untuk meminimalkan efek karena bias dan kesalahan *drift*, dan digunakan untuk menghitung orientasi *body* atau *object frame* (OF) dengan perambatan Quaternion mulai dari orientasi yang diperkirakan pada langkah sebelumnya. Pengukuran *accelerometer* dan *magnetometer* digabungkan dengan parameter konstanta pembobotan β melalui algoritma *gradient descent*, yang formulasinya dilaporkan dalam keluaran dari algoritma *gradient descent* kemudian digunakan untuk memperbaiki orientasi yang diperkirakan dengan hanya mempertimbangkan pengukuran *gyroscope*. Alur kerja dari metode Madgwick pada (Gambar 3.3), yaitu untuk mengetahui representasi pergeseran *frame* berdasarkan sumbu garis normal gravitasi bumi menggunakan Persamaan *gradient descents*. Apabila di *cross product* ketiga Quaternion tersebut menghasilkan matrik yang baru.

3.5 Perhitungan Posisi dan Orientasi

Posisi atau perpindahan diketahui berdasarkan data percepatan, data quaternion dan matrix rotasi yang didapatkan dari filter madgwick. Data percepatan (m/s^2) dari sensor *accelerometer* pada MPU 9250 yang diintegrasikan sebanyak dua kali sehingga menghasilkan data perpindahan (m). Diagram blok untuk perhitungan posisi dapat dilihat pada (Gambar 3.4).



Gambar 3. 4 Diagram Blok Perhitungan Posisi

Berdasarkan (Gambar 3.4) madgwick filter berfungsi untuk mendapatkan nilai quaternion berdasarkan data percepatan, kecepatan sudut dan medan magnet dari MPU 9250. Nilai quaternion yang didapatkan kemudian diubah menjadi matrix rotasi untuk kemudian dikali dengan nilai percepatan dari sensor *accelerometer* yang sudah diproses sehingga memiliki satuan m/s^2 .

3.5.1 Konversi Percepatan Menjadi Referensi Bumi

Sensor IMU pada buoy mendapatkan data percepatan, kecepatan sudut dan medan magnet pada referensi sensor itu sendiri (*body frame*). Sehingga untuk mengukur perpindahan perlu untuk merubah referensinya menjadi referensi bumi (*earth frame*). Konversi ini dilakukan dengan cara mengalikan data dengan matrix rotasi, data yang dikalikan adalah data percepatan (m/s^2) karena data percepatan ini yang digunakan secara langsung diintegrasikan untuk mendapatkan nilai perpindahan [42]. Perhitungan konversi dilakukan berdasarkan Persamaan (3.1).

$$a_{(gs)} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = a_{(bs)} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \cdot R_{(q)} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \quad (3.1)$$

Dimana $a_{(gs)}$ adalah nilai akselerasi pada referensi bumi, $a_{(bs)}$ adalah nilai percepatan pada referensi sensor itu sendiri dan $R_{(q)}$ adalah rotasi matrix dalam quaternion dengan q_0 sebagai bilangan *real* dan q_1, q_2, q_3 sebagai bilangan kompleks [29][42][43]. Nilai akselerasi baik dalam referensi bumi dan referensi sensor dalam bentuk matrix 3x1 sedangkan matrix rotasi dalam bentuk matrix 3x3. Matrix rotasi dapat dilihat pada Persamaan (3.2) berikut.

$$R_{(q)} = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_0q_2 + q_1q_3) \\ 2(q_1q_2 + q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_0q_1 + q_2q_3) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \quad (3.2)$$

Matrix rotasi pada Persamaan (3.2) yang digunakan pada skripsi ini untuk merubah referensi alat menjadi referensi Bumi, perhitungannya dilakukan pada aplikasi matlab dengan masukan data yang sudah di ambil sebelumnya.

3.5.2 Konversi Percepatan Menjadi Perpindahan

Data akselerasi 3 sumbu (x, y, z) dalam bentuk matrix 3x1 yang sudah diubah menjadi referensi bumi harus diintegral kan untuk mendapatkan data perpindahan. Integral pertama menghasilkan data kecepatan (m/s) dari sebelumnya (m/s^2). Data kecepatan harus diintegrasikan kembali untuk menghasilkan data perpindahan (m). Dasar proses ini bisa dilakukan berdasarkan Persamaan (3.3) dan (3.4).

$$\Delta v = dT \cdot a \quad (3.3)$$

$$\Delta s = dT \cdot \Delta v \quad (3.4)$$

Lambang v adalah kecepatan dalam vektor, a adalah percepatan dalam bentuk vektor, s adalah perpindahan dalam bentuk vektor dan dT adalah *interval* waktu. Persamaan tidak bisa dilakukan secara *real time*, sedangkan penelitian ini membutuhkan data perpindahan secara *real time*, sehingga dipilihlah metode *trapezoidal integration*. Integral metode trapezoidal menggunakan data saat ini dan data pengukuran sebelumnya dalam perhitungannya [16]. Persamaan integral metode trapezoidal bisa dilihat pada Persamaan.

$$y(n) = y(n - 1) + \frac{1}{2} \cdot dT \cdot [x(n - 1) + x(n)], n = 1, 2, 3, \dots \quad (3.5)$$

Pada Persamaan (3.5) $y(n)$ adalah hasil dari integral, $y(n-1)$ adalah hasil sebelumnya, dT adalah selisih waktu pengukuran dari sensor, $x(n-1)$ adalah masukan sebelumnya dan $x(n)$ adalah masukan baru atau masukan saat ini. Metode ini lebih akurat dari metode integral biasa pada Persamaan dan hanya perlu menyimpan satu data sebelumnya.

3.6 Perhitungan Gelombang Laut

Gelombang adalah gangguan dari media fluida melalui mana energi dipindahkan. Gelombang laut bergerak pada antarmuka antara lautan dan atmosfer. Gelombang disebabkan oleh gesekan antara angin dan permukaan air, gaya tarik gravitasi, gempa bumi atau letusan gunung berapi [4][17].

3.6.1 Tinggi Gelombang

Tinggi gelombang adalah perbedaan antara ketinggian permukaan maksimum dan minimum air dalam gelombang yang relevan [19]. Tinggi gelombang (H) dihitung menggunakan algoritma ZCD dengan data masukan adalah data perpindahan vertikal dari alat hasil integrasi ganda dari percepatan vertikal *earth frame*. Tinggi gelombang dihitung dari jarak tegak lurus puncak dan lembah gelombang.

3.6.2 Periode Gelombang

Periode gelombang adalah jarak antar dua puncak atau lembah gelombang. Periode gelombang (T_p) diperoleh dengan mengukur waktu antara puncak berurutan [4][17]. Perhitungan perioda gelombang dilakukan menggunakan algoritma ZCD.