Sistem Odometri Robot Bawah Air Berbasis *Encoder Thruster Controller* (ETC) untuk Estimasi Posisi dan Jarak Tempuh

Buku Manual Penggunaan Aplikasi

Ditulis Oleh

1. Ryan Satria Wijaya
2. Dhaniel Beny Wardhana
3. Wildan Mahfudh Khoirul Murtadho
4. Aditia Ummardiansyah
5. Aldi Ali Yudianto
6. Adifta Justiano
7. Krisna Dwi Cahya
8. Rendy Androleo Rizaldy
9. Rizki Agustino

Politeknik Negeri Batam

2025

Kata Pengantar

Puji syukur kami panjatkan ke hadirat Tuhan Yang Maha Esa atas limpahan rahmat, karunia, dan petunjuk-Nya sehingga buku manual penggunaan aplikasi Sistem Odometri Robot Bawah Air Berbasis *Encoder Thruster Controller* untuk Estimasi Posisi dan Jarak Tempuh ini dapat disusun dan diselesaikan dengan baik. Buku manual ini disusun sebagai panduan bagi pengguna maupun pengembang dalam memahami, menginstalasi, mengoperasikan, serta memelihara sistem odometri yang diterapkan pada robot bawah air.

Sistem odometri yang dibahas dalam manual ini dirancang khusus untuk mengatasi keterbatasan sistem navigasi konvensional yang tidak dapat berfungsi optimal di lingkungan bawah air, seperti GPS. Melalui pemanfaatan sensor *encoder* yang terpasang pada poros thruster, sistem mampu menghitung jumlah putaran baling-baling secara akurat, yang kemudian dikonversikan menjadi data jarak tempuh, sehingga sistem dapat menghitung posisi robot secara *real-time* dengan tingkat ketelitian yang baik. Dengan pendekatan ini, robot bawah air dapat melakukan navigasi, pemetaan area, menjaga stabilitas posisi saat manuver, serta mendukung pengumpulan data penelitian di lingkungan akuatik secara efisien.

Penyusunan manual ini diharapkan dapat memberikan penjelasan teknis secara detail dan sistematis, mulai dari proses instalasi perangkat lunak, perakitan perangkat keras, konfigurasi sistem, hingga penjelasan algoritma pengolahan data odometri. Selain itu, manual ini juga memuat contoh kode program, diagram sistem, serta dokumentasi pendukung lain yang memudahkan pengguna dalam menerapkan sistem odometri ini secara praktis di lapangan.

Kami menyampaikan penghargaan dan ucapan terima kasih yang sebesar-besarnya kepada seluruh tim pengembang, pembimbing, serta pihak-pihak yang telah memberikan kontribusi dalam pengembangan sistem ini, baik dari aspek akademik, teknis, maupun pengujian di lapangan. Semoga buku manual ini dapat bermanfaat, tidak hanya sebagai panduan operasional, tetapi juga sebagai referensi pengembangan teknologi robotika bawah air di masa yang akan datang.

Batam, 13 Juni 2025

Ryan Satria Wijaya

(Tim Pengembang)

Daftar Isi

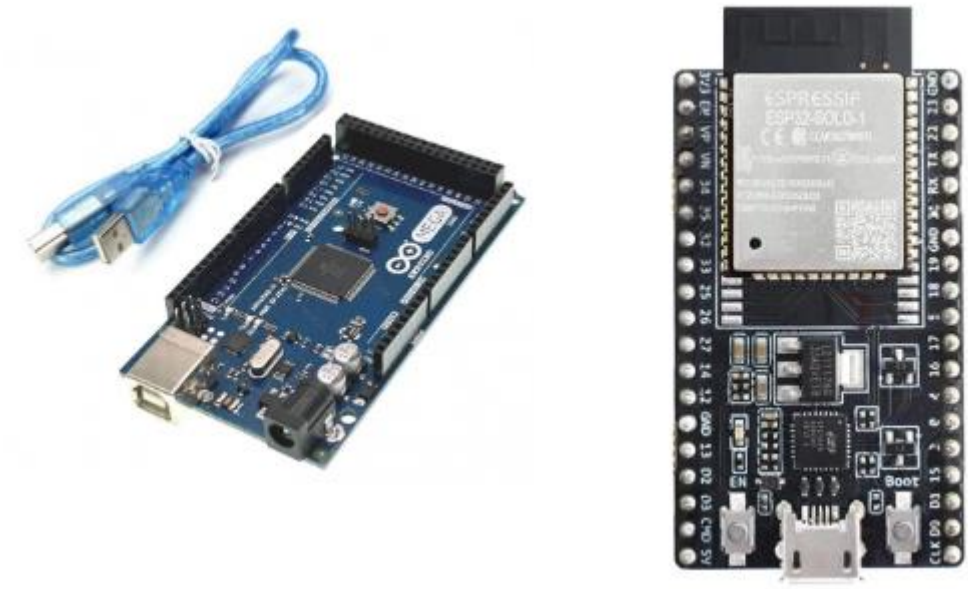
Daftar Gambar

1. Instalasi Aplikasi

1.1. **Hardwire**

Sistem Odometri Robot Bawah Air memerlukan beberapa komponen dan *mikrokontroller* untuk dapat bekerja, seperti berikut :

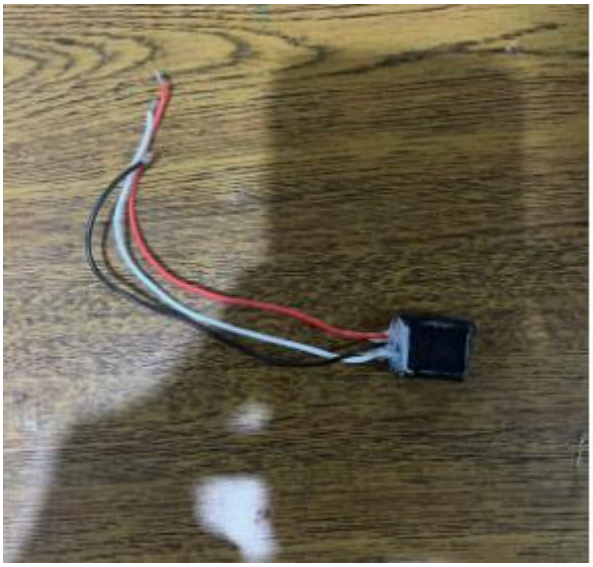
1. Arduino/ESP32
   Fungsi mikrokontroler dalam sistem ini adalah sebagai pusat pengolah data yang bertugas membaca sinyal pulsa dari encoder thruster controller untuk menghitung jarak tempuh, serta melakukan perhitungan estimasi posisi robot bawah air secara *real-time* menggunakan algoritma odometri. Selain itu, mikrokontroler juga mengatur kecepatan thruster melalui sinyal PWM, menyimpan data pergerakan untuk keperluan logging, dan mengirimkan hasil pengukuran ke sistem monitoring melalui komunikasi serial atau nirkabel. Keseluruhan proses dikendalikan secara otomatis dan kontinu untuk mendukung navigasi presisi di lingkungan bawah air.



Gambar 1. Mikrokontroller Arduino dan ESP32.

2. Modul encoder thruster controller
   Modul encoder thruster controller berfungsi untuk membaca jumlah putaran thruster melalui sinyal dari sensor encoder yang terpasang pada poros motor thruster. Setiap putaran motor menghasilkan sejumlah pulsa yang dikirim ke mikrokontroler untuk dihitung sebagai jarak tempuh.

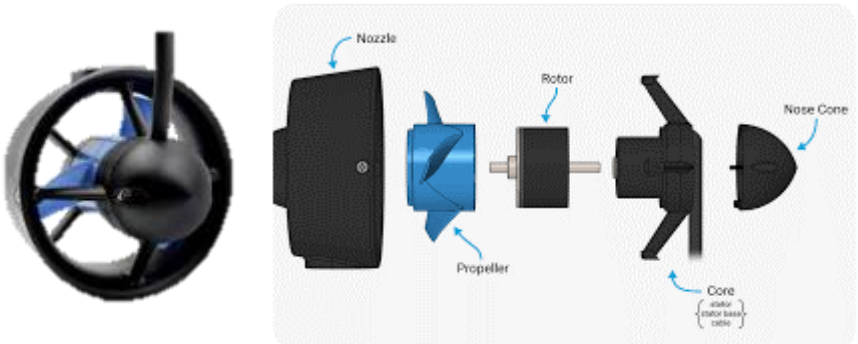
Cara kerja:
saat thruster berputar, encoder menghasilkan pulsa dengan mendeteksi magnet yang ada pada rotor thruster T200, kemudian data pulsa ini dihitung oleh mikrokontroler sebagai estimasi posisi dari robot bawah air.



Gambar 2. Encoder Thruster Controller.

3. *Thruster* T200

Thruster T200 berfungsi menghasilkan gaya dorong yang menyebabkan pergerakan robot bawah air. Saat *thruster* berputar, poros motor berputar sehingga encoder yang terpasang pada dekat poros tersebut bisa mengambil data magnet permanen pada rotor thruster. Setiap putaran poros menghasilkan sinyal pulsa yang dibaca oleh *encoder thruster controller*. Dari sinyal ini, *encoder thruster controller* dapat menghitung jumlah putaran motor, yang kemudian dikonversikan menjadi jarak tempuh. Dengan kata lain, gerakan fisik dari *thruster* menjadi sumber utama data jarak, yang diolah oleh sistem odometri untuk memperkirakan posisi robot secara *real-time*.



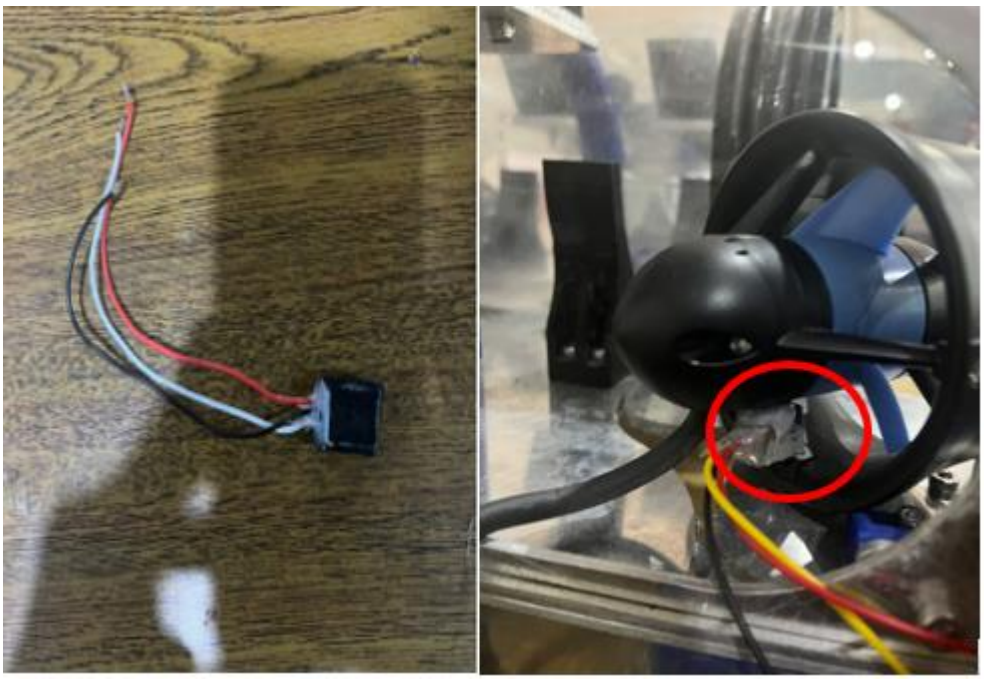
Gambar 3 Thruster T200 blue robotic.

1.2. Instalasi Arduino IDE

Sistem Odometri Robot Bawah Air membutuhkan *platform* pemrograman untuk mengembangkan program kendali mikrokontroler, yaitu menggunakan Arduino IDE. Arduino IDE digunakan untuk menulis, mengunggah, serta memantau program yang berjalan pada mikrokontroler (misalnya ESP32, Arduino UNO, atau sejenisnya).Unduh Arduino IDE dari situs resmi:
https://www.arduino.cc/en/software

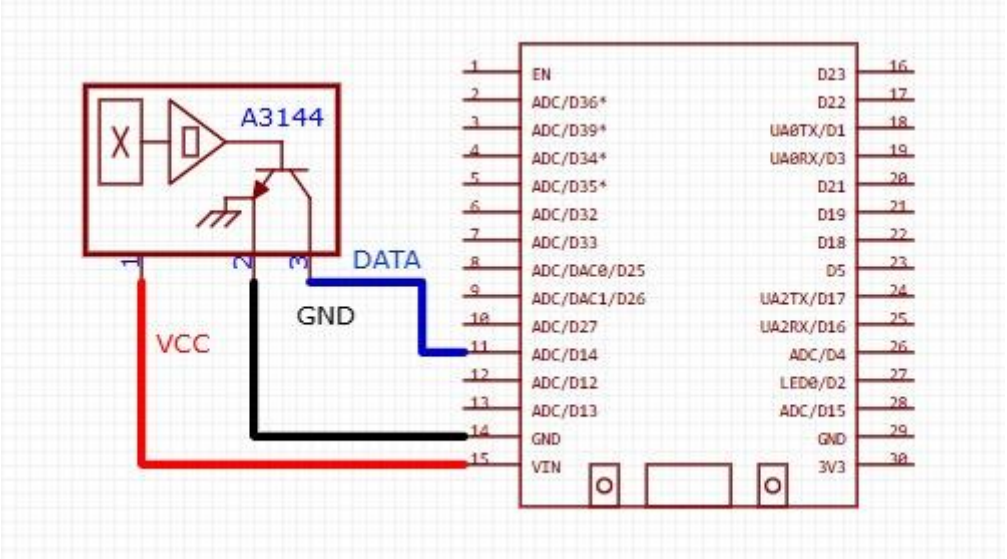
2. Pengoperasian Aplikasi

1. Pasangkan modul ETC pada *thruster*.

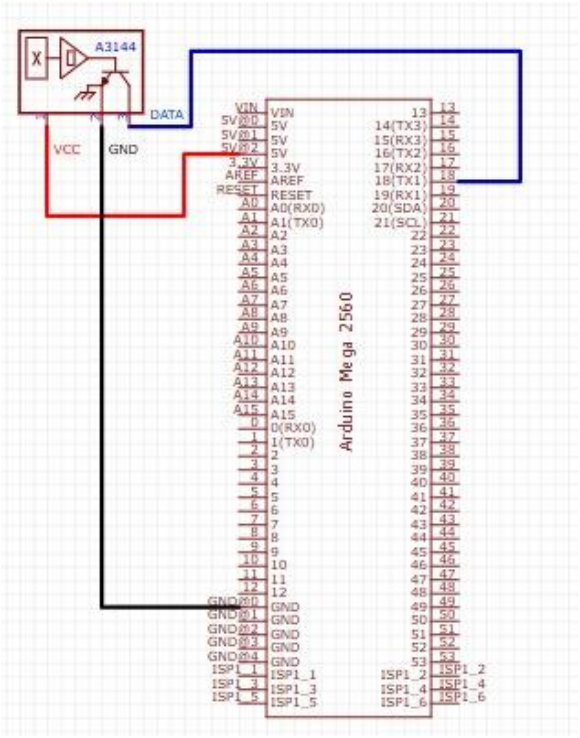


Gambar 4. Posisi pemasangan encoder thruster controller.

2. Hubungkan perangkat modul ETC pada *mikrokontroller* (Arduino,ESP32,dll). Pin Data ETC harus dihubungkan ke pin interrupt pada *mikrokontroller,* pin interrupt pada *mikrokontroller* Arduino mega ada pada pin 2,3,18,19,20,21.

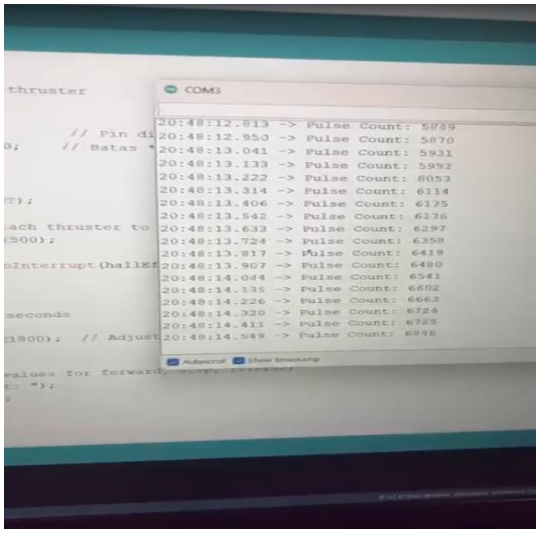


Gambar 5. Wiring elektrikal ETC dan ESP32.



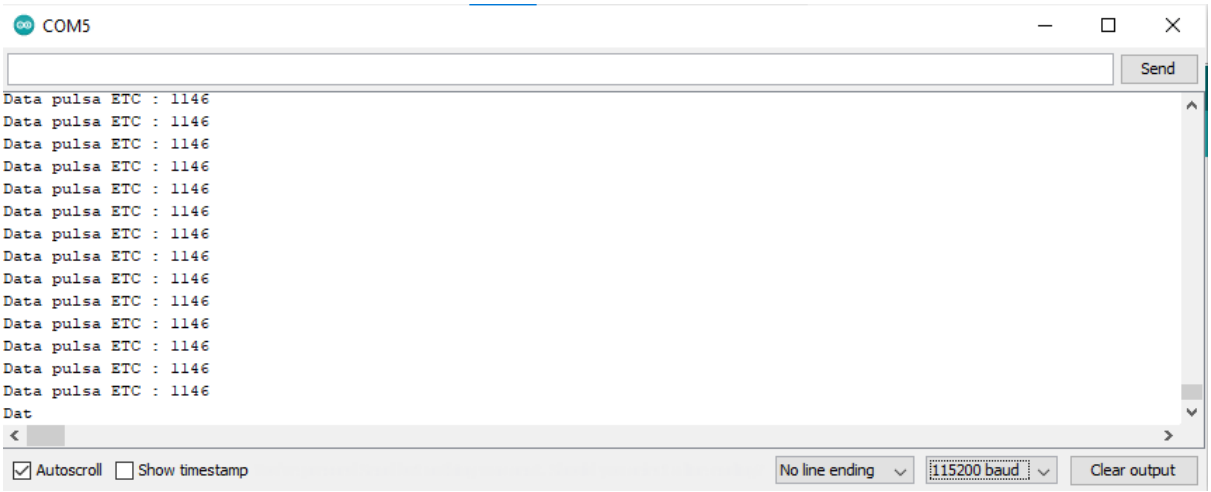
Gambar 6. Wiring elektrikal ETC dan arduino mega.

3. Lakukan percobaan pengambilan data pulsa dari ETC dengan menjalankan thruster untuk memastikan bahwa perangkat sudah dapat berfungsi, jika data pulsa belum masuk coba pastikan kembali cara pemasangan perangkat sehingga bisa mendeteksi magnet yang ada pada thruster. Caranya setelah kita apload program kalibrasi, selanjutnya kita. Berikut link video proses percobaan cek pulsa dari ETC dengan menyalakan *thruster* :
   https://youtube.com/shorts/YhA4GcbtSS4?feature=share

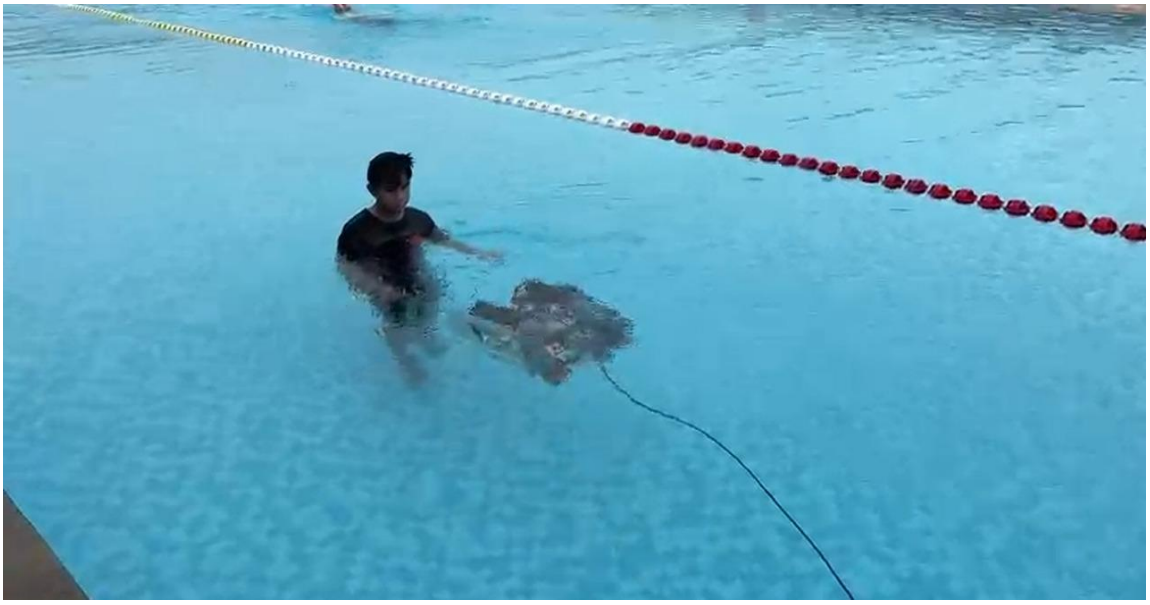


Gambar 7. Hasil pengecekan data pulsa dari ETC.

4. Lakukan kalibrasi data untuk mendapatkan data pulsa dalam meter. Caranya ukur terlebih dahulu area kalibrasi dengan meteran dalam skala 1 meter dan beri penanda, selanjutnya jalankan robot bawah air sesuai jalur area kalibrasi kemudian kita lihat data pulsa etc pada jarak-jarak yang sudah di tandai sebelumnya. Berikut contoh link proses kalibrasi data :
https://youtube.com/shorts/d8QRDUAUJZY?feature=share



Gambar 8. Tampilan serial monitor saat kalibrasi data.



Gambar 9. Proses pengambilan data kalibrasi.

Jarak (Meter)	Data pulsa ETC
0	0
1	1000
2	2000
3	3000
4	4000
5	5000
6	6000
7	7000
8	8000
9	9000
10	10000

Gambar 10. Contoh hasil data kalibrasi.

5. Masukkan hasil data kalibrasi ke program utama ETC

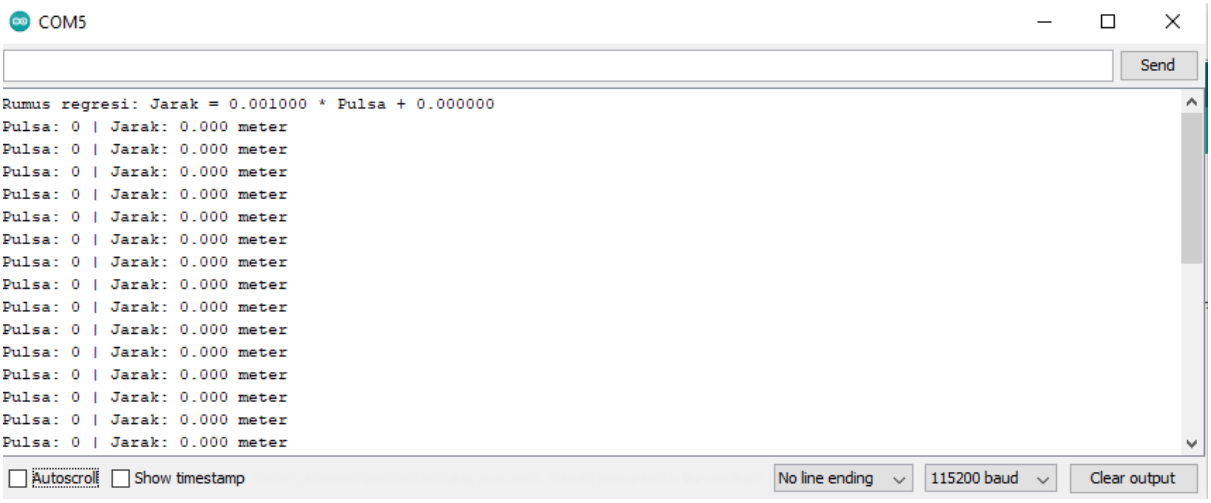
```
int encoderPin = 2;
int pulseCount = 0;

// Data kalibrasi: isi data pulsa hasil pengukuran pada masing-masing jarak
const int N = 11;
float jarakMeter[N] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10};  // Jarak (meter)
float pulsa[N] = {0, 1000, 2000, 3000, 4000, 5000, 6000, 7000, 8000, 9000, 10000};  // Masukkan data pulsa hasil pengukuran

// Variabel regresi
float a = 0.0;
float b = 0.0;
```

Gambar 11. Memasukkan data kalibrasi pada program utama.

6. Jalankan robot bawah air dan lihat hasil navigasi jarak melalui serial monitor pada Arduino ide.



Gambar 12. Tampilan serial monitor pada program utama.

3. *Source Code* Aplikasi

3.1. Program ETC

Aplikasi **Sistem Odometri Robot Bawah Air Berbasis *Encoder Thruster Controller* (ETC) untuk Estimasi Posisi dan Jarak Tempuh** dapat diunduh pada tautan berikut ini https://github.com/barelangmrt-robotics/Sistem-Odometri-Robot-Bawah-Air-Berbasis-Encoder-Thruster-Controller-ETC-untuk-Estimasi-Posisi

3.2. Skrip program kalibrasi Perangkat

Skrip program kalibrasi **Sistem Odometri Robot Bawah Air Berbasis *Encoder Thruster Controller* (ETC)** adalah sebagai berikut

```
const int etcPin1 = 2; // Pin untuk sensor 1

volatile int ETC1 = 0; // Jumlah pulsa untuk sensor 1



void setup() {

  // Mulai komunikasi serial

  Serial.begin(115200);



  // Atur pin sensor sebagai input

  pinMode(etcPin1, INPUT_PULLUP);



  // Pasang interrupt untuk menghitung pulsa dari setiap sensor

  attachInterrupt(digitalPinToInterrupt(etcPin1), halleffect1, RISING);

}



void loop() {

  // Tampilkan jumlah ETC yang terdeteksi dari setiap sensor

    Serial.print("Data pulsa ETC : ");

  Serial.println(ETC1);
```

```
}


// Fungsi interrupt untuk menghitung ETC dari sensor 1
void halleffect1() {
  ETC1++;
}
```

3.3. Skrip program Utama ETC

Skrip program utama **Sistem Odometri Robot Bawah Air Berbasis *Encoder Thruster Controller* (ETC)** adalah sebagai berikut

```
// Deklarasi pin encoder
int encoderPin = 2;


// Variabel untuk menghitung jumlah pulsa dari sensor
int pulseCount = 0;


// Data kalibrasi: data hasil pengukuran jarak dan pulsa
const int N = 11;  // Jumlah data kalibrasi
float jarakMeter[N] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10};  // Data jarak dalam meter
float pulsa[N] = {0, 1000, 2000, 3000, 4000, 5000, 6000, 7000, 8000, 9000, 10000};  // Data pulsa sesuai pengukuran


// Variabel hasil perhitungan regresi linear
float a = 0.0;  // Slope (gradien)
float b = 0.0;  // Intercept (titik potong sumbu Y)
```

```
// Fungsi setup, dijalankan sekali saat pertama kali menyala
void setup() {
  Serial.begin(115200); // Memulai komunikasi serial dengan baudrate 9600

  // Mengatur pin encoder sebagai input pull-up
  pinMode(encoderPin, INPUT_PULLUP);

  // Menghubungkan interrupt pada rising edge ke fungsi countPulse()
  attachInterrupt(digitalPinToInterrupt(encoderPin), countPulse, RISING);

  // Memanggil fungsi untuk menghitung persamaan regresi dari data kalibrasi
  hitungRegresi();

  // Menampilkan hasil persamaan regresi
  Serial.print("Rumus regresi: Jarak = ");
  Serial.print(a, 6);
  Serial.print(" * Pulsa + ");
  Serial.println(b, 6);
}

// Fungsi loop, berjalan berulang-ulang selama sistem aktif
void loop() {
  // Mengamankan pembacaan variabel pulseCount dari interrupt
  noInterrupts();  // Menonaktifkan interrupt sementara
  long pulses = pulseCount;  // Membaca nilai sementara
```

```
  interrupts();  // Mengaktifkan kembali interrupt

  // Menghitung jarak berdasarkan hasil regresi
  float distance = a * pulses + b;

  // Menampilkan hasil pembacaan pulsa dan jarak
  Serial.print("Pulsa: ");
  Serial.print(pulses);
  Serial.print(" | Jarak: ");
  Serial.print(distance, 3);
  Serial.println(" meter");

  delay(500);  // Menunggu 500 ms sebelum pembacaan berikutnya
}

// Fungsi interrupt: setiap ada pulsa baru, pulseCount bertambah 1
void countPulse() {
  pulseCount++;
}

// Fungsi untuk menghitung koefisien regresi linear berdasarkan data kalibrasi
void hitungRegresi() {
  float sumX = 0.0;
  float sumY = 0.0;
  float sumXY = 0.0;
```

```
  float sumX2 = 0.0;


  // Melakukan perhitungan sigma-sigma untuk regresi linear
  for (int i = 0; i < N; i++) {
    sumX += pulsa[i];
    sumY += jarakMeter[i];
    sumXY += pulsa[i] * jarakMeter[i];
    sumX2 += pulsa[i] * pulsa[i];
  }


  // Rumus regresi linear sederhana: Y = aX + b
  a = (N * sumXY - sumX * sumY) / (N * sumX2 - sumX * sumX);
  b = (sumY - a * sumX) / N;
}
```