

DAFTAR TABEL

Tabel 4.1 Pengujian memulai aplikasi dan komunikasi <i>roboboat</i>	32
Tabel 4.2 Pengujian memberikan misi pada <i>roboboat</i> berupa beberapa <i>waypoints</i> ...	33
Tabel 4.3 Pengujian penampilan data log ketika misi telah selesai.....	33
Tabel 4.4 Nilai pembacaan servo 90^0	35
Tabel 4.5 Nilai pembacaan servo 120^0	36
Tabel 4.6 Nilai pembacaan servo 60^0	36
Tabel 4.7 Hasil pengujian waypoint misi pertama.....	42
Tabel 4.8 Hasil Inputan jalur waypoint	45

DAFTAR RUMUS

Persamaan (1) rumus untuk menghitung jarak	6
Persamaan (2) rentang sudut azimuth	7
Persamaan (3) rentang sudut pole	7
Persamaan (4) representasi sudut azimuth pada arah sebenarnya	7
Persamaan (5) representasi sudut pole pada arah sebenarnya	7
Persamaan (6) Rumus mencari sudut azimuth.....	7
Persamaan (7) Rumus mencari sudut azimuth.....	7
Persamaan (8) Rumus mencari sudut azimuth.....	7
Persamaan (9) Rumus mencari sudut azimuth.....	7
Persamaan (10) Rumus Bearing.....	8
Persamaan (11) Rumus Bearing off	9
Persamaan (12) Rumus MSE	47

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Pada saat ini perkembangan sistem *autonomous* sangat berkembang pesat salah satunya sudah digunakan pada jenis USV (*Unmanned Surface Vehicle*). Salah satu jenis dari USV ialah *roboboat*. *Roboboat* dapat diimplementasikan untuk memonitoring daerah perairan yang tidak terjangkau manusia, *aerial mapping*, membawa muatan dan pantauan perairan. Tetapi masih terdapat kekurangan dalam hal memonitoring daerah perairan yang tidak terjangkau oleh manusia dalam hal ini pada umumnya *roboboat* masih dikendalikan oleh seorang pilot menggunakan sebuah *remote control* sehingga seorang pilot harus *standby* untuk mengetahui pergerakan dari *roboboat*.

Oleh karena itu, dibuatlah sebuah sistem yang memungkinkan *roboboat* untuk bergerak secara *autonomous* dengan menggunakan sistem navigasi yang menggunakan algoritma *waypoint*. Jadi *roboboat* dapat bergerak secara *autonomous* dengan dikendalikan secara otomatis menggunakan PC (*Personal Computer*) sebagai antar muka pengguna dimana komunikasinya menggunakan modul *wireless RF 433 MHz*, tanpa menggunakan *remote control* dan bisa mengikuti jalur yang telah dibuat dengan bantuan perangkat GPS (*Global Positioning System*) yang telah terintegrasi dalam *roboboat*. Perangkat GPS akan menangkap sinyal NMEA dari satelit GPS yang menghasilkan koordinat *latitude* dan *longitude* terhadap lokasi perangkat GPS tersebut berada.

Roboboat diprogram dengan koordinat *waypoint* yang dipilih dan akan melaju dengan kecepatan konstan pada saat menuju *waypoint*. Namun pengguna masih dapat mengontrol *roboboat* apabila *roboboat* mengalami masalah sistem dengan bantuan dari *remote control*

1.2 Rumusan Masalah

Pada tugas akhir ini terdapat beberapa permasalahan yang muncul, diantaranya :

- a. Bagaimana merancang suatu sistem mekanik dan elektronik pada *roboboat* ?
- b. Bagaimana mengetahui hasil dari sistem navigasi pada *roboboat* sehingga dapat bergerak secara *autonomous* pada jalur yang telah dibuat ?
- c. Bagaimana cara merancang perangkat lunak sebagai antar muka pengguna dan *roboboat* ?

1.3 Batasan Masalah

Batasan –batasan masalah yang terdapat dalam tugas akhir ini adalah :

- a. Pengendali utama untuk *Roboboat* ialah Laptop
- b. Menggunakan Ardupilot Mega 2560 sebagai unit pemrosesan data
- c. Pengiriman data menggunakan RF433 MHz.
- d. Penggunaan *roboboat* yang dirancang hanya untuk diluar ruangan
- e. Durasi waktu pakai *roboboat* kurang lebih 30 Menit
- f. Roboboat berupa prototype dari sebuah kapal dengan dimensi kurang lebih 70 cm x 40 cm
- g. Tidak memperhatikan delay sistem secara menyeluruh
- h. GPS yang di gunakan adalah Neo-7M
- i. Tidak memperhatikan Kontrol PID secara menyeluruh

1.4 Tujuan Penelitian

- a. Merancang sistem mekanik dan elektronik pada *roboboat*
- b. Merancang perangkat lunak sebagai antar muka pengguna dengan *roboboat*
- c. Mengetahui dan menganalisa hasil error dari sistem navigasi *roboboat* yang dapat bergerak secara *autonomous* dengan mengambil sampel dari perbandingan jalur yang telah dibuat dan dilalui *roboboat*

- d. Mengetahui estimasi jarak tempuh maksimum *roboboat* berdasarkan lama waktu laju sistem *roboboat*

1.5 Metoda Penelitian

Langkah yang akan ditempuh dalam menyelesaikan tugas akhir ini diantaranya adalah :

1. Tahap Studi Literatur

Pada tahap ini dilakukan pencarian dan pengumpulan literatur-literatur dan kajian-kajian yang berkaitan dengan masalah-masalah yang ada pada Tugas akhir ini, baik berupa artikel, buku referensi, jurnal, internet, dan sumber-sumber lain yang berhubungan dengan masalah Tugas akhir.

2. Tahap Perancangan dan Realisasi Alat

Pada tahap ini setelah mempelajari literatur yang ada dilakukan pembuatan rancangan-rancangan yang kemudian direalisasikan rancangan tersebut ke dalam suatu rangkaian dan perangkat.

3. Tahap pengujian Sistem

Pada tahap selanjutnya berdasarkan standar yang ada, tahap selanjutnya adalah melakukan pengujian sistem untuk melihat kinerja sistem tersebut.

1.6 Jadwal Pelaksanaan

Berikut jadwal pelaksanaan penelitian perancangan dan implementasi autonomous roboboat dengan sistem navigasi GPS yang terintegrasi pada android. Sesuai dengan Surat Ketetapan dari Universitas Telkom yakni waktu pengerjaan penelitian Tugas Akhir selama satu semester (waktu normal 6 bulan), maka diestimasikan waktu pengerjaan adalah 5 bulan terhitung mulai bulan Januari 2016

No.	Kegiatan	Bulan 1	Bulan 2	Bulan 3	Bulan 4	Bulan 5
1	Studi Literatur					
2	Analisis Masalah					
3	Perancangan dan Simulasi					
4	Implementasi					
5	Eksperimen dan Analisis					
6	Penyusunan Laporan					

BAB II DASAR TEORI

2.1 Unmanned autonomous vessel (USV)

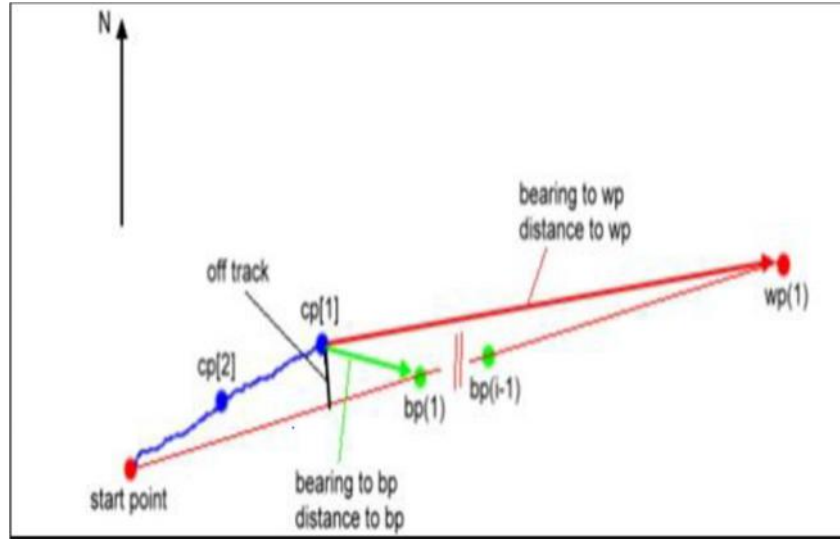
Unmanned Surface Vehicle (USV) adalah kendaraan tanpa awak yang umumnya beroperasi di perairan ^[1] USV dikendalikan jarak jauh oleh seorang navigator atau secara otomatis dengan prinsip dasar hampir sama dengan kendaraan di darat. Untuk teknologi penunjang sistem kendali dan komunikasi data, antara UAV dan USV kurang lebih banyak persamaan.^[6] Karena prinsip utama dari USV adalah *autonomous* jadi sistem yang digunakan pun harus benar-benar teruji karena kalau sistem yang digunakan tidak berjalan optimal USV akan sulit sekali untuk dilacak keberadaannya.

2.2 Waypoint Navigation Algorithm

Pada tugas akhir saya *waypoints* merupakan dasar yang paling penting dalam proses sistem navigasi. *Waypoints* adalah set koordinat yang mengidentifikasi titik dalam ruang fisik. Koordinat yang digunakan dapat bervariasi tergantung pada aplikasi. Untuk navigasi darat koordinat ini dapat mencakup bujur dan lintang. Untuk navigasi udara termasuk ketinggian *waypoints* dapat berkembang secara luas untuk digunakan oleh orang awam karena pengembangan sistem navigasi canggih, seperti (*Global Positioning System*) GPS. dan jenis-jenis tertentu navigasi radio. *Waypoints* terletak di permukaan bumi biasanya didefinisikan dalam dua dimensi (misalnya, bujur dan lintang); yang digunakan di atmosfer Bumi atau di luar angkasa didefinisikan dalam setidaknya tiga dimensi (empat jika waktu adalah salah satu koordinat, karena mungkin untuk beberapa titik arah luar Bumi).^[2]

^[1]**Basing point** adalah posisi data GPS yang berdasarkan data yang diberikan oleh pengguna sesuai jalur untuk menghubungkan 2 *waypoint* dan digunakan untuk mengurangi *roboboat* keluar dari jalur. **Initial distance / jarak awal** merupakan jarak dari posisi awal benda ke arah *waypoint* pertama yang aktif, jarak awal tidak mungkin berubah selama sistem navigasi berlangsung hingga mencapai *waypoint* tercapai dan *waypoint* yang berikutnya diaktifkan **Current point / titik saat ini** : merupakan sebuah string data GPS yang memiliki format NMEA dari mana posisi saat *roboboat*

diaktifkan, dan digunakan untuk menghitung dan memperbaiki arah saat *roboboat* berjalan **Jarak inisialisasi** : adalah jarak antara 2 waypoint. **Initial Bearing** : adalah jarak yang ditempuh apabila *roboboat* tidak sesuai dengan jalur untuk mencapai waypoint selanjutnya



Gambar 2.1 Istilah- istilah yang terdapat pada Navigasi waypoint

Prinsip utama dari sistem navigasi *waypoint* ialah *roboboat* menerima data dari GPS yang meliputi lintang bujur untuk posisi awal. dan pengguna memberikan daftar *waypoints* yang ingin *roboboat* capai. Dengan menghitung posisi saat ini sebagai proses awal dari sistem navigasi kemudian *roboboat* akan melintasi rute yang telah ditentukan pada *waypoint* berikutnya Untuk perhitungan yang lebih tepat dari jarak antara dua tempat yang jauh, rumus jarak lingkaran besar dapat digunakan rumus yang menggunakan sifat trigonometri dari suatu koordinat - termasuk sinus, cosinus, arccosinus - dan perkiraan jari-jari Bumi, maka jarak dari suatu tempat dapat dihitung.^[1]

$$\text{Dist} = 2 \sin^{-1} \left[\cos(\text{lata}) \cdot \cos(\text{latb}) \cdot \left(\sin \left(\frac{\text{lon}a - \text{lon}b}{2} \right) \right)^2 + \left(\sin \left(\frac{\text{lata} - \text{latb}}{2} \right) \right)^2 \right] \quad (1)$$

Lat a = Latitude of the waypoint "a"

Lat b = Latitude of the waypoint "b"

Lon a = Longitude of the waypoint "a"

Lon b = Longitude of the waypoint "b"

2.2.1 Azimuth dan Pole angle

Azimuth adalah besar sudut antara utara sebenarnya dengan titik target. Pada subbab sebelumnya sudah dijelaskan bagaimana cara menghitung jarak berdasarkan *longitude* dan *latitude*. Namun untuk mengetahui posisi *roboboat* sesungguhnya kita perlu mengkonversi koordinat *latitude* dan *longitude* kedalam nilai azimuth dan *pole angle*, untuk menyempurnakan jarak maka persamaan sudut azimuth (φ) dan *pole angle* (θ) adalah^[12]

$$-180^\circ \leq \varphi \leq 180 \quad (2)$$

$$\text{dan } -90^\circ \leq \theta \leq 90^\circ \quad (3)$$

$$\theta = \begin{array}{l} - \text{lat untuk ara}^\circ = \text{selatan} \\ + \text{lat untuk ara}^\circ = \text{utara} \end{array} \quad (4)$$

$$\varphi = \begin{array}{l} - \text{lon untuk ara}^\circ = \text{timur} \\ + \text{lon untuk ara}^\circ = \text{barat} \end{array} \quad (5)$$

Untuk nilai latitude dan longitude pada bumi, dengan persamaan ini setiap posisi pada bumi dapat dijelaskan dengan ^[12]

$$f: \begin{array}{l} r \\ \text{lon a} \\ \text{lata} \end{array} \rightarrow \begin{array}{l} r \cdot \cos \text{lona} \cdot \sin \text{lata} \\ r \cdot \sin \text{lona} \cdot \sin(\text{lata}) \\ r \cdot \cos(\text{lata}) \end{array} = \begin{array}{l} x \\ y \\ z \end{array} \quad (6)$$

$$\cos \alpha = \frac{\cos \text{lona} \cdot \sin \text{lata}}{\sin \text{lona} \cdot \sin(\text{lata})} \cdot \frac{\cos \text{lonb} \cdot \sin \text{latb}}{\sin \text{lonb} \cdot \sin(\text{latb})} \quad (7)$$

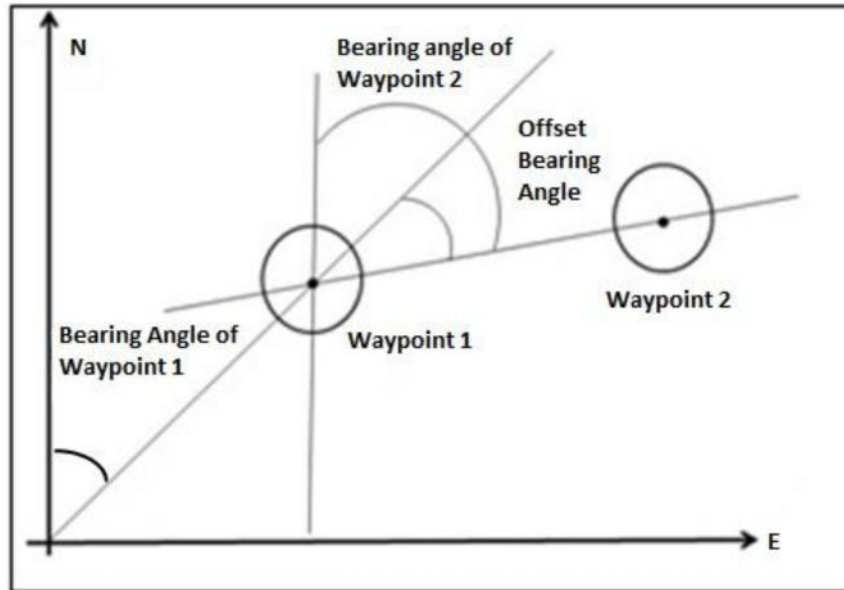
$$= \cos(\text{lat}_a) \cdot \cos(\text{lat}_b) + \sin(\text{lat}_a) \cdot \sin(\text{lat}_b) \cdot (\cos(\text{lon}_a) \cdot \cos(\text{lon}_b) + \sin(\text{lon}_a) \cdot \sin(\text{lon}_b)) \quad (8)$$

Dan sudut (α) dapat dengan mudah dihitung dengan cara merubah fungsi arcos menjadi arctan pada persamaan ini sehingga menjadi ^[12]

$$\arccos(\alpha) = \arctan\left(\frac{-\alpha}{\sqrt{1-\alpha^2}}\right) + 2 \cdot \arctan(1) \quad (9)$$

2.2.2 GPS Bearing dan off Bearing

GPS Bearing (Degree)



Gambar 2.2 Contoh yang menunjukkan metode untuk memperoleh *Offset bearing*

$$\text{bearing} = \cos^{-1} \frac{(\sin(\text{lat}_b) - \sin(\text{lat}_a) \cdot \cos(\text{dist}))}{\sin(\text{dist}) \cdot \cos(\text{lat}_a)} \quad (10)$$

Lat a = Latitude of the waypoint "a"

Lat b = Latitude of the waypoint "b"

Lon a = Longitude of the waypoint "a"

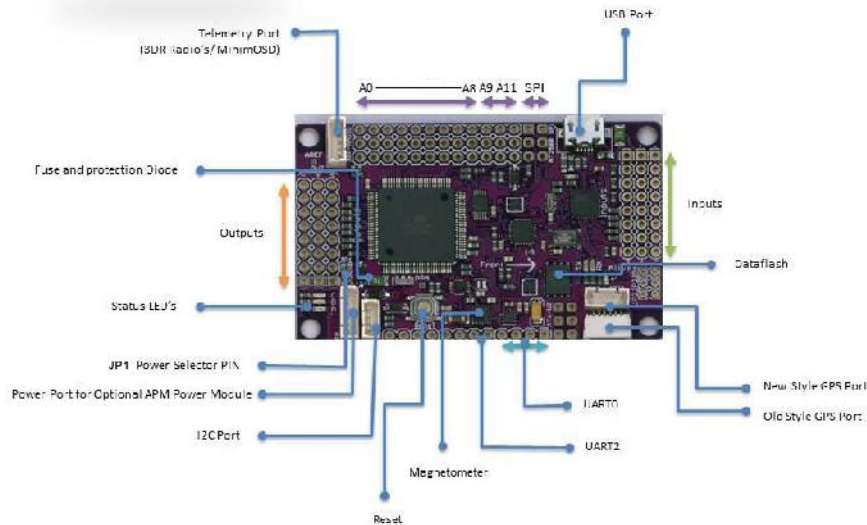
Lon b = Longitude of the waypoint "b"

Pada algoritma navigasi menghitung perbedaan sudut selama melewati *waypoints*. Sudut awal adalah sudut antara dua titik arah yang diperoleh dari persamaan diatas. Jarak dan sudut awal tidak berubah kecuali jika roboboat telah sampai pada

waypoint selanjutnya. Sudut menuju *waypoint* adalah sudut dari posisi sekarang dari *roboboat* ke *waypoint* selanjutnya, sama seperti sudut ke posisi sebenarnya ('bearing to bp') adalah sudut dari jarak sekarang ke posisi sebenarnya. Sudut-sudut ini dan jarak yang sebenarnya berubah diantara 2 *waypoint* dan digunakan untuk menghitung kesalahan sudut dengan mengambil perbedaan dari sudut posisi sekarang.

$$\text{bearing}_{\text{off}} = \begin{cases} \text{bearing}_{\text{to_wp}} - \text{bearing}_{\text{current}} & \text{for } \text{dist}_{\text{to_wp}} \leq \text{dist}_{\text{to_bp}} \\ \text{bearing}_{\text{to_wp}} - \text{bearing}_{\text{current}} & \text{for } \text{dist}_{\text{to_wp}} > \text{dist}_{\text{to_bp}} \end{cases} \quad (11)$$

2.3 Ardupilot Mega 2560



Gambar 2.2 Ardupilot Mega 2560 Board Features
(<http://ardupilot.org/copter/docs/common-25-and-26-overview.html>)

Ardupilot Mega merupakan sistem autopilot dengan sumber terbuka yang sangat lengkap artinya kita dapat mengembangkan sistem yang sudah ada pada Ardupilot Mega. Pada tugas akhir saya Ardupilot digunakan mengontrol gerak dari roboboat yang sudah dilengkapi dengan sistem *autonomous*. Komponen ini bisa digunakan sebagai pusat kontrol untuk *fixed wing*, *multirotor*, *rover*, *boat* dan semua wahana yang membutuhkan kemampuan *autonomous*. Pada Ardupilot bisa juga dipasang komponen tambahan seperti GPS untuk keperluan *waypoint*.

Selain itu fitur – fitur lain yang terdapat di dalam *Ardupilot* ini yaitu :

- Pin input/output dapat digunakan sesuai dengan kebutuhan pengguna
- 3-axis gyro MPU 6000, accelerometer 6Dof dan magnetometer
- Digital kompas HMC5883L
- Barometer MS5611-01BA03
- IC yang digunakan yaitu Atmel's ATMEGA 2560-16AU dan ATMEGA32U-2.

Dengan menggunakan IC atme's ATMEGA 2560 dan ATMEGA32U-2. Memungkinkan kita untuk mengembangkan sistem yang ada dengan menggunakan arduino yang disambungkan ke ardupilot. Catu daya yang diperlukan untuk menjalankan ardupilot ialah 5 Volt.^[8]

2.4 Global Positioning System (GPS)

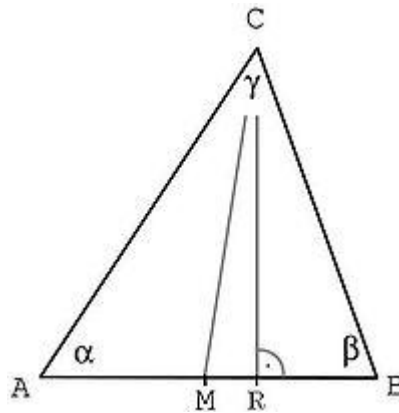


Gambar 2.3 GPS Neo -7M

(<http://www.banggood.com/CRIUS-NEO-GPS-MAG-V2-NEO-7M-Modul10-With-Compass-p-951898.html>)

Dasar navigasi berbasis GPS (Global Positioning System) adalah *triangulation* dari sebuah satellit. *Triangulation* adalah proses dalam mencari titik koordinat dan

jarak dari sebuah titik dengan mengukur sudut antara titik tersebut dan dua titik referensi lainnya yang sudah diketahui posisi dan jarak antara keduanya. Untuk melakukan proses *triangulation*, penerima dari GPS mengukur jarak menggunakan waktu tempuh signal radio dari satellite kepada pengguna. Untuk mengukur waktu tempuh, GPS membutuhkan waktu yang akurat dari *feedback* yang diberikan oleh signal radio. Pada umumnya, arsitektur dari sistem GPS sama dengan sistem *satellite* lainnya. Arsitektur terbagi menjadi 3 bagian yaitu *space segment*, *ground segment*, dan *user segment*. *Space segment* adalah komponen yang terdapat di luar angkasa. *Ground segment* berfungsi dalam manajemen satellite dalam memantau dan mengendalikan GPS pada operasi orbit. Sedangkan *user segment* berfungsi untuk mempermudah pengguna dalam mengendalikan GPS.^[9]



Gambar 2.4 Metode Pengukuran Triangulasi

(<http://planologiunisba575859.blogspot.co.id/2014/12/metode-pengukuran-triangulasi.html>)

$$l = \frac{d}{\tan \alpha} + \frac{d}{\tan \beta} \quad (12)$$

$$d = l \left(\frac{1}{\tan \alpha} + \frac{1}{\tan \beta} \right) \quad (13)$$

$$d = \frac{l \sin \alpha \sin \beta}{\sin(\alpha + \beta)}$$

Persamaan (12), (13),(14) merupakan persamaan yang menjelaskan cara kerja dari proses penentuan lokasi yang terdapat pada *Global Positioning System* ⁽¹⁴⁾

Pada tugas akhir ini saya menggunakan GPS Neo -7M yg di dalamnya sudah memiliki modul kompas HMC5883L. Dan memiliki beberapa spesifikasi diantaranya ialah

Tipe penerima:

- 1.50 kanal, GPS L1 frequency, C/A Code. SBAS: WAAS, EGNOS, MSAS
2. Sensitivitas penjejak & navigasi: -161 dBm (reakuisisi dari blank-spot: -160 dBm)
3. Sensitivitas start: -139 dBm pada cold-start, -155 dBm pada hot start
4. Kecepatan pembaharuan data / navigation update rate: 10 Hz
5. Akurasi penetapan lokasi GPS secara horisontal: 2,5 meter (SBAS = 2m)
6. Rentang frekuensi pulsa waktu yang dapat disetel: 0,25 Hz hingga 10 MHz
7. Akurasi sinyal pulsa waktu: RMS 30 ns (99% dalam kurang dari 60 ns) dengan granularitas 21 ns atau 15 ns saat terkompensasi
8. Akurasi kecepatan: 0,1 meter / detik
9. Akurasi arah (heading accuracy): 0,5°
10. Menggunakan baud rate sebesar 9600

2.5 Electronic Speed Controller (ESC)

Electronic Speed Controller (ESC) yang terdapat pada *Controller board* berfungsi untuk mengendalikan putaran motor *brushless* agar robot dapat bergerak sesuai dengan nilai referensi yang diberikan, input dari ESC ini merupakan nilai PWM yang dibangkitkan oleh *Main controller board*. [11]