



PROYEK AKHIR - VE180626

**PERANCANGAN DAN PENGATURAN KECEPATAN MOTOR
BRUSHLESS DC SEBAGAI MODUL PEMBELAJARAN**

Muchamad Rizki Abdillah
NRP 10311500010028

Dosen Pembimbing
Ir. Joko Susila. MT.
Ciptian Weried Priananda S.ST. MT.

DEPARTEMEN TEKNIK ELEKTRO OTOMASI
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya 2020



FINAL PROJECT - VE180626

**DESIGN BRUSHLESS DC MOTOR SPEED SETTING AS A
LEARNING MODULE**

Muchamad Rizki Abdillah
NRP 10311500010028

Supervisor
Ir. Joko Susila MT.
Ciptian Weried Priananda S.ST. MT

*DEPARTMENT ELECTRICAL AND AUTOMATION ENGINEERING
Vocational Faculty
Intitut Teknologi Sepuluh November
Surabaya 2020*

-----Halaman ini sengaja dikosongkan-----

PERNYATAAN KEASLIAN PROYEK AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Proyek Akhir saya dengan judul “**Perancangan dan Pengaturan Kecepatan Motor *Brushless* DC Sebagai Modul Pembelajaran**” adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 12 Juni 2020

Muchamad Rizki Abdillah
NRP 10311500010028

-----Halaman ini sengaja dikosongkan-----

PERANCANGAN DAN PENGATURAN KECEPATAN MOTOR BRUSHLESS DC SEBAGAI MODUL PEMBELAJARAN

PROYEK AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Memperoleh Gelar Ahli Madya Teknik
Pada
Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember

Menyetujui :

Pembimbing 1

Pembimbing 2



Ciptian Weried Priananda
21/8/2020

Ir. Joko Susila, MT.
NIP. 19621005 199003 1-003

Ciptian Weried Priananda S.ST. MT.
NPP. 1990201711060

SURABAYA
JUNI, 2020

-----Halaman ini sengaja dikosongkan-----

PERANCANGAN DAN PENGATURAN KECEPATAN MOTOR BRUSHLESS DC SEBAGAI MODUL PEMBELAJARAN

Nama : Muchamad Rizki Abdillah
NRP : 10311500010028
Pembimbing 1 : Ir. Joko Susila, MT.
Pembimbing 2 : Ciptian Weried Priananda S.ST. MT.

ABSTRAK

Dalam proyek akhir ini, untuk mengetahui kecepatan dari motor *brushless* DC diperlukan sebuah modul yang bisa memberikan data yang tepat yang dapat dipahami dengan mudah. Selain itu, untuk mengontrol kecepatan dari motor *brushless* DC diperlukan pula sebuah modul kontrol. Namun, modul khusus untuk kontrol kecepatan motor *brushless* DC yang menampilkan hasil dengan Grafik tidak mudah untuk dibuat.

Oleh karena itu untuk mempermudah proyek akhir ini, diperlukan suatu modul yang praktis dan efisien. Sehingga muncul ide untuk membuat suatu modul yang digunakan untuk mengontrol kecepatan motor *brushless* DC dengan mikrokontroler. Modul ini terdiri dari 3 komponen utama, yaitu arduino sebagai mikrokontroler, motor *brushless* DC dan sensor kecepatan.

Pada Proyek Akhir ini, dihasilkan bahwa kontrol kecepatan motor *brushless* DC ini menggunakan metode PWM yang di kendalikan dengan penentuan nilai PWM untuk mendapatkan nilai kecepatan motor dan Arus *brushless* DC yang ditampilkan melalui tampilan grafik.

Kata Kunci : Motor *Brushless* DC, PWM, Arduino, Kecepatan, Arus.

-----Halaman ini sengaja dikosongkan-----

DESIGN BRUSHLESS DC MOTOR SETTING AS LEARNING MODULE

Name : Muchmad Rizki Abdillah

NOR : 10311500010028

Supervisor 1: Ir. Joko Susilo, MT.

Supervisor 2: Ciptian Wieried Priyananda S.ST. MT.

ABSTRACT

In this final project, we need a module that can provide the right data to find out the speed of a brushless motor DC be seen directly. In addition, to control the speed of a DC brushless motor, a control module is also needed. However, special modules for DC brushless motor speed control are not easy to make.

Therefore to facilitate this final project, a module that is practical and efficient is needed. So the idea arises to make a module that is used to control the brush-less DC motor speed with a microcontroller. This module consists of 3 main components, namely Arduino as a microcontroller, DC brushless motor and speed sensor.

In this final project, it is produced that the speed control of this brushless DC motor uses the PWM method which is controlled by determining the PWM value to get the value of the motor speed and brushless DC current which is displayed through a graphical display.

Keywords : *Brushless DC Motor, PWM, Arduino, Speed, Current.*

-----Halaman ini sengaja dikosongkan-----

KATA PENGANTAR

Puji syukur kami panjatkan ke hadirat Tuhan Yang Maha Esa karena atas limpahan rahmat dan karunia-Nya, penulis dapat menyelesaikan Proyek Akhir ini dengan judul :

“PERENCANAAN DAN PENGATURAN KECEPATAN MOTOR BRUSHLESS DC SEBAGAI MODUL PEMBELAJARAN”

Proyek Akhir ini merupakan sebagian syarat untuk menyelesaikan mata kuliah dan memperoleh nilai pada Proyek Akhir.

Dengan selesainya Proyek Akhir ini penulis menyampaikan terima kasih sebesar-besarnya kepada:

1. Orang Tua serta keluarga besar atas limpahan doa, kasih sayang, dukungan dan dorongan baik berupa moral atau materil bagi penulis.
2. Bapak Imam Arifin. ST., MT. selaku Kepala Departemen Teknik Elektro Otomasi, FV-ITS Surabaya.
3. Bapak Ir.Joko Susila MT. selaku Dosen Pembimbing.
4. Seluruh staf pengajar dan administrasi Departemen Teknik Elektro Otomasi FV-ITS.
5. Seluruh Mahasiswa Teknik Elektro Otomasi khususnya angkatan 2015.
6. Semua pihak yang telah banyak membantu untuk menyelesaikan Proyek Akhir ini yang tidak dapat penulis sebutkan satu persatu.

Penulis menyadari masih banyak kekurangan dalam Proyek Akhir ini. Kritik dan saran untuk perbaikan tugas ini sangat diperlukan. Akhir kata semoga tugas ini dapat bermanfaat bagi kita semua.

Surabaya, 12 Juni 2020

Penulis

-----Halaman ini sengaja dikosongkan-----

DAFTAR ISI

HALAMAN JUDUL	i
PERNYATAAN KEASLIAN PROYEK AKHIR	v
LEMBAR PENGESAHAN	vii
ABSTRAK	ix
ABSTRACT	xi
KATA PENGANTAR	xiii
DAFTAR ISI	xv
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan Dan Manfaat	1
1.3 Permasalahan	2
1.4 Batasan Masalah	2
1.5 Sistematika Penulisan	2
1.6 Relevansi	3
BAB II TEORI PENUNJANG	4
2.1 Motor <i>Brushless</i> DC [1]	5
2.1.1 Cara Kerja Motor <i>Brushless</i> DC	6
2.1.2 Konstruksi Motor <i>Brushless</i> DC	7
2.1.3 ESC (<i>Electronic Speed Controller</i>)	9
2.1.4 Prinsip Kerja Motor <i>Brushless</i> DC	7
2.2 Sensor <i>Rotary Encoder</i>	8
2.3 Mikrokontroler Arduino Mega 2560	9
2.3.1 Cara Kerja Motor <i>Brushless</i> DC	11
BAB III PERANCANGAN DAN PEMBUATAN ALAT	13
3.1 Blok Fungsional Sistem	15
3.2 Perancangan Perangkat Keras (<i>Hardware</i>)	17
3.2.1 Perancangan Mekanik	17
3.2.1.1 Motor <i>Brushless</i> DC	17
3.2.1.2 <i>Electronic Speed Control</i> (ESC)	17
3.2.1.3 Flywheel	15
3.2.2 Perancangan Elektrik	19
3.2.2.1 Rangkaian Sensor <i>Rotary Encoder</i>	19

3.2.2.2 Rangkaian Arduino Mega 2560	19
3.3. Perancangan Perangkat Lunak (<i>Software</i>)	21
3.3.1 Pemrograman <i>Software</i> Arduino	22
BAB IV PENGUJIAN DAN ANALISA	24
4.1 Pengujian Sensor Rotary Encoder.....	22
4.2 Pengujian Kecepatan Putaran Motor Beushless DC dan Arus tanpa Beban.....	265
4.3 Pengujian Kecepatan Putaran Motor Beushless DC dan Arus dengan Beban.....	29
BAB V PENUTUP	151
5.1. Kesimpulan	35
5.2 Saran	35
DAFTAR PUSTAKA.....	37
LAMPIRAN A	A-1
LAMPIRAN B.....	B-5
LAMPIRAN C	C-1
RIWAYAT PENULIS	C-3

DAFTAR GAMBAR

Gambar 2.1	Contoh BLDC.....	5
Gambar 2.2	Skema Kerja Motor Brushless DC	6
Gambar 2.3	<i>Outrunner Construction</i>	7
Gambar 2.4	<i>Inrunner Construction</i>	7
Gambar 2.5	ESC.....	9
Gambar 2.6	<i>Rotary Encoder</i>	10
Gambar 2.7	Arduino Mega 2560	10
Gambar 2.8	<i>Setting Serial Port</i>	13
Gambar 3.1	Blok Fungsional Perancangan	14
Gambar 3.2	Blok Diagram Sistem <i>Close Loop</i>	14
Gambar 3.3	Beban Flywheel tampak samping	20
Gambar 3.4	Beban Flywheel tampak depan	20
Gambar 3.5	Ilustrasi bentuk Flywheel.....	20
Gambar 3.6	Rangkaian Rotary Encoder	20
Gambar 3.7	Rangkaian Arduino Mega 2560.....	20
Gambar 3.8	Skematik Arduino Mega 2560 dan Komponen	20
Gambar 3.9	Flowchart Arduino Mega 2560.....	20
Gambar 4.1	Realisasi Plant Motor BLDC	220
Gambar 4.2	Grafik Linier Kalibrasi Sensor Kecepatan	23
Gambar 4.3	Grafik Linier Kalibrasi Sensor Arus.....	24
Gambar 4.4	Pengujian Kecepatan Motor dan Arus Brushless Dc tanpa beban dengan nilai Duty Cycle 10%	26
Gambar 4.5	Pengujian Kecepatan Motor dan Arus Brushless Dc tanpa beban dengan nilai Duty Cycle 10%	26
Gambar 4.6	Pengujian Kecepatan Motor dan Arus Brushless Dc dengan beban dengan nilai Duty Cycle 30%	28
Gambar 4.7	Pengujian Kecepatan Motor dan Arus Brushless Dc dengan beban dengan nilai Duty Cycle 10%	29

-----Halaman ini sengaja dikosongkan-----

DAFTAR TABEL

Tabel 2.1 Spesifikasi Motor <i>Brushless</i> DC Tipe <i>Outrunner</i> RC Timer A2212 1400KV	6
Tabel 2.2 Deskripsi Arduino Mega 2560	10
Tabel 4.1 Pengujian Sensor <i>Rotary Encoder</i>	24
Tabel 4.2 Pengujian Kecepatan dan Arus Motor BLDC tanpa Beban 26	
Tabel 4.3 Pengujian Kecepatan dan Arus Motor BLDC dengan Beban	28

-----Halaman ini sengaja dikosongkan-----

BAB I

PENDAHULUAN

1.1. Latar Belakang

Seiring dengan perkembangan zaman, kebutuhan akan motor listrik untuk kendaraan listrik yang memiliki efisiensi tinggi, torsi yang tinggi, kecepatan yang tinggi dan dapat divariasikan, dan biaya perawatan yang rendah sangat amat tinggi. Hanya saja motor yang digunakan secara umum saat ini, yakni motor *Direct Current* (DC), motor DC ini memiliki biaya perawatan yang tinggi. Biaya perawatan tinggi ini muncul akibat kerusakan pada brush dalam komutasi motor DC. Sedangkan untuk brushless dc ini memiliki komponen stator yang terbuat dari kumparan dan rotor yang terbuat dari permanen magnet. Motor *Brushless Direct Current* (BLDC) adalah alternatif pengganti motor DC. Sesuai namanya motor BLDC tidak memiliki sikat pada komutatornya. Sebagai gantinya digunakan komutator elektrik. Motor ini biasanya banyak digunakan dalam *automotive, aerospace, medical*, industri, dan instrumentasi.

Pada dunia industri biasanya menggunakan pengontrol yang banyak dan kompleks. Dengan menggunakan arduino tidak perlu perangkat *chip* programmer karena di dalamnya sudah ada *bootloader* yang akan menangani upload program dari komputer. Hal ini bertujuan mempermudah pengguna mengetahui hasil pengamatan yang telah mereka lakukan. Untuk dapat memahami itu maka diperlukannya sebuah pembelajaran yang mendukung menguasai penggunaan motor BLDC dan kontroler.

Untuk menghadapi Industri 4.0 maka diperlukannya sebuah persiapan mental dan ilmu materi baik secara pendidikan formal maupun non-formal. Pembelajaran ini diharapkan dapat menjadi bekal persiapan meningkatkan sumber daya manusia pada dunia industri yang saat ini sangat maju.

1.2. Tujuan Dan Manfaat

Tujuan kami menuliskan Proyek Akhir ini adalah:

1. Merancang kontrol kecepatan motor *brushless* DC.
2. Mengimplementasikan kontrol kecepatan motor *brushless* DC.

Harapan dirancangnya alat ini adalah agar alat ini dapat membantu sebagai metode pembelajaran yang berhubungan dengan kontrol kecepatan.

1.3. Permasalahan

Permasalahan yang diangkat pada Proyek Akhir ini adalah belum adanya modul pembelajaran praktis guna mempermudah proses pembelajaran mengenai kontrol kecepatan motor *brushless* DC.

1.4. Batasan Masalah

Pembatasan pada Proyek Akhir ini meliputi :

1. Arah putaran motor *brushless* DC tidak diperhatikan.
2. Hanya mengatur kecepatan motor *brushless* DC menggunakan *driver* ESC 30 Ampere.
3. Menggunakan Motor Generator DC 12 Volt.
4. Motor Brushless DC tipe A2212 1400KV.
5. Hanya menggunakan satu Beban Pengereman

1.5. Sistematika Penulisan

Dari proses pembuatan alat Proyek Akhir ini dimulai dari studi literatur, menentukan gambar desain alat, membuat alat, membuat program, menguji alat secara keseluruhan, analisa data, serta dapat menyusun laporan akhir dengan sistematika penulisan yaitu pendahuluan, teori penunjang, perancangan alat, pengukuran dan analisa serta penutup.

BAB I PENDAHULUAN

Berisi latar belakang pembuatan alat, permasalahan, pembatasan masalah, tujuan, metodologi, sistematika penulisan serta manfaat Proyek Akhir.

BAB II TEORI PENUNJANG

Menjelaskan teori yang berisi teori-teori penunjang yang dijadikan landasan prinsip dasar dan mendukung dalam perencanaan dan pembuatan alat yang dibuat.

BAB III PERANCANGAN ALAT

Membahas tentang tahap-tahap perancangan mekanik dan perancangan sistem kontrolernya.

BAB IV PENGUKURAN DAN ANALISA

Membahas tentang pengukuran dan pengujian alat atau rangkaian yang digunakan pada Proyek Akhir ini.

BAB V PENUTUP

Berisikan tentang kesimpulan atas hasil yang diperoleh serta saran-saran atas kekurangan dan kelemahan Proyek Akhir ini.

1.6. Relevansi

Membuat modul pembelajaran praktis yang mudah digunakan sebagai sarana pembelajaran kontrol kecepatan motor *brushless* DC.

-----Halaman ini sengaja dikosongkan-----

BAB II

TEORI PENUNJANG

Sistem kendali atau sistem kontrol adalah suatu alat (kumpulan alat) untuk mengendalikan, memerintah, dan mengatur keadaan dari suatu sistem. Banyak contoh dalam bidang ah satunya adalah sistem kendali/kontrol kecepatan pada motor *brushless* DC.

Pada bab ini akan dibahas mengenai industri/instrumentasi dan dalam kehidupan kita sehari-hari di mana sistem ini dipakai, salah satu teori yang berkaitan dengan peralatan yang akan dirancang. Teori yang mendukung penyelesaian Proyek Akhir ini diantaranya adalah mengenai : motor *brushless* dc dengan mikrokontroler Arduino UNO REV3.

2.1. Motor *Brushless* DC [1]

Salah satu jenis motor listrik yang paling banyak digunakan akhir-akhir ini adalah motor *brushless* DC dimana motor DC ini tidak menggunakan *brush* (sikat) untuk proses komutasi. Motor *brushless* DC sangat cocok untuk diaplikasikan pada produk yang menuntut reliabilitas dan efisiensi yang tinggi. Secara umum, dapat dikatakan bahwa motor *brushless* DC dapat menghasilkan torsi yang besar dan mempunyai *range* RPM yang tinggi. Motor *brushless* DC merupakan salah satu jenis motor sinkron dimana medan magnet yang dihasilkan oleh stator dan medan magnet yang dihasilkan oleh rotor berputar pada frekuensi yang sama. Untuk lebih jelasnya mengenai motor ini dapat dilihat pada Gambar 2.1 dan Tabel 2.1.

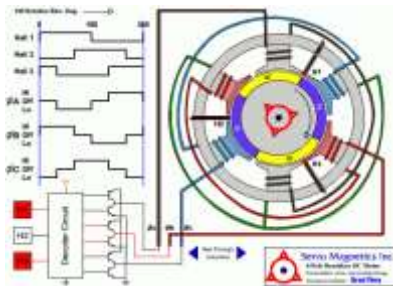


Gambar 2. 1 Contoh BLDC

**Tabel 2. 1 Spesifikasi Motor *Brushless* DC Tipe *Outrunner* RC
Timer A2212 1400KV [2]**

No.	Parameter	Nilai	
1.	Berat Motor	47 gram	
2.	KV	1400 rpm/Volt	
3.	Tegangan	Tegangan Minimal	7 Volt
		Tegangan Maksimal	15 Volt
4.	<i>Input</i> Arus	Arus Nominal	4 Ampere
		Arus Maksimal	18 Ampere
5.	<i>Maximal Power</i>	210 Watt	
6.	<i>Input</i> Baterai	Lithium Polimer 2S-4S	

2.1.1 Cara Kerja Motor *Brushless* DC [2]



Gambar 2. 2 Skema Kerja Motor *Brushless* DC [2]

Cara kerja pada motor *brushless* DC cukup sederhana, yaitu magnet yang berada pada poros motor akan tertarik dan terdorong oleh gaya elektromagnetik yang diatur oleh *Electronic Speed Kontroller* (ESC). Hal ini yang membedakan motor *bushless* DC dengan motor DC dimana motor DC menggunakan sikat mekanis yang berada pada komutator untuk mengatur waktu dan memberikan medan magnet pada lilitan. Motor *brushless* DC ini juga berbeda dengan motor AC yang pada umumnya menggunakan siklus tenaga sendiri untuk mengatur waktu dan memberi daya pada lilitan. Motor *brushless* DC dapat memberikan rasio daya dan beban yang lebih tinggi secara signifikan dan memberikan efisiensi yang lebih baik dibandingkan motor tanpa sikat tradisional.

2.1.2 Konstruksi Motor *Brushless* DC [1]

Motor BLDC terdiri dari 2 jenis yaitu *outrunner* dan *inrunner*. *Outrunner* merupakan jenis motor BLDC yang memiliki stator terletak di dalam motor dan rotor terletak diluar sehingga bagian dari motor yang berputar merupakan bagian luar dari motor. Sedangkan *inrunner* merupakan jenis motor yang memiliki stator di bagian luar motor dan rotor berada pada bagian dalam motor sehingga bagian motor yang berputar adalah bagian dalam motor.



Gambar 2. 3 *Outrunner* Construction [1]



Gambar 2. 4 *Inrunner* Construction [1]

Setiap motor *brushless* DC memiliki dua bagian utama, rotor (bagian berputar) dan stator (bagian stasioner). Bagian penting lainnya dari motor adalah sebagai berikut :

1. Rotor

Rotor adalah bagian pada motor yang berputar karena adanya gaya elektromagnetik dari stator, dimana pada motor *brushless* DC bagian rotornya berbeda dengan rotor pada motor DC konvensional yang hanya tersusun dari satu buah elektromagnet yang berada diantara *brushes* (sikat) yang terhubung pada dua buah motor hingga delapan pasang kutub magnet permanen berbentuk persegi panjang yang saling direkatkan menggunakan semacam “*epoxy*” dan tidak ada *brushes*-nya.

2. Stator

Stator adalah bagian pada motor yang diam/statis dimana fungsinya adalah sebagai medan putar motor untuk memberikan gaya elektromagnetik pada rotor sehingga motor dapat berputar. Pada motor *brushless* DC statornya terdiri dari 12 belitan (elektromagnet) yang bekerja secara elektromagnetik dimana statornya terhubung dengan tiga buah kabel untuk disambungkan pada rangkaian kontrol sedangkan pada motor DC konvensional statornya terdiri dari dua buah kutub magnet permanen. Belitan stator pada motor *brushless* DC terdiri dari dua jenis, yaitu belitan stator jenis *trapezoidal* dan jenis *sinusoidal*. Yang menjadi dasar perbedaan kedua jenis belitan stator tersebut terletak pada hubungan antara koil dan belitan stator yang bertujuan untuk memberikan EMF (*Electro Motive Force*) balik yang berbeda.

Berikut ini adalah bagian penting pada stator :

a. *Axle*

Axle atau sumbu adalah batang yang berfungsi sebagai sumbu putar motor, terpusat pada rotor dan dirangkai bersama rotor.

b. *Sensor Hall*

Tidak seperti motor DC, *brushed* komutasi dari motor *brushless* DC diatur secara elektronik agar motor dapat berputar, stator harus di-*energize* secara berurutan dan teratur. Sensor *hall* inilah yang berperan dalam mendeteksi pada bagian rotor mana yang ter-*energize* oleh fluks magnet sehingga proses komutasi yang berbeda (enam langkah komutasi) dapat dilakukan oleh stator dengan tepat karena sensor *hall* ini dipasang menempel pada stator. *Hall sensor* ini ditempatkan setiap 120° pada jarak antar kutub stator hal ini bertujuan agar deteksi terhadap *vector* fluks stator yang dihasilkan akurat setiap perpindahan komutasi, arus yang mengalir tetap terjaga konstan pada setiap fasa. Prinsip kerja *hall sensor* sendiri membutuhkan arus yang mengalir terus jika ingin digunakan sebagai pendeteksi fluks magnet.

2.1.3 ESC (*Electronic Speed Controller*) [1]



Gambar 2. 5 Electronic Speed Controller

ESC merupakan rangkaian elektronik yang berfungsi sebagai pengatur kecepatan putaran motor pada motor RC (*Radio Control*), cara kerjanya yaitu dengan cara menterjemahkan sinyal yang diterima *receiver* dari *transmitter*. Di pasaran terdapat berbagai *merk* ESC dengan kekuatan arus (*current rating*) dan kekuatan Voltase (*voltage rating*) serta *feature* yang ditawarkan.

Penjelasan yang cepat mengenai ESC adalah ESC harus secara akurat menghubungkan dan memutuskan koneksi antara 3 masukan *input* dan 3 belitan pada stator agar rotor dapat berputar.

Untuk menentukan ESC yang akan kita gunakan sangatlah penting untuk mengetahui kekuatan (*peak current*) dari motor. Pilihlah ESC yang kekuatannya melebihi kekuatan motor. Misalnya, dari data kita dapatkan kekuatan motor adalah 12A (sesuai dengan *datasheet* motor) pada saat *throttle* terbuka penuh. Sebaiknya ESC yang akan kita gunakan adalah ESC yang berkekuatan 30A. Jika kita paksakan menggunakan ESC 10A kemungkinan pada saat *throttle* dibuka penuh, ESC akan panas bahkan terbakar.

2.1.4 Prinsip Kerja Motor *Brushless* DC [1]

Motor BLDC ini dapat bekerja ketika stator yang terbuat dari kumparan diberikan arus 3 fasa. Akibat arus yang melewati kumparan pada stator timbul magnet :

$$B = \frac{\mu Ni}{2l} \dots \dots \dots (2.1)$$

Dimana N merupakan jumlah lilitan, I merupakan arus, l merupakan Panjang lilitan dan μ merupakan permeabilitas bahan.

Karena arus yang diberikan berupa arus DC 3 fasa sinusoidal, nilai medan magnet dan polarisasi setiap kumparan akan berubah-ubah setiap saat.

Akibat yang ditimbulkan dari adanya perubahan polarisasi dan besar magnet tiap kumparan adalah terciptanya medan putar magnet dengan kecepatan yang tertera pada persamaan berikut ini :

$$n_s = \frac{120f}{p} \dots\dots\dots(2.2)$$

Dimana f merupakan frekuensi arus *input* dan p merupakan jumlah *pole* stator. Sedangkan untuk mendapatkan total daya (*power*) yang dihasilkan, maka persamaannya adalah :

$$W = V \times A \dots\dots\dots(2.3)$$

Jadi sebenarnya W adalah daya (*power*) yang dihasilkan oleh tegangan dan arus listrik untuk menggerakkan *propeller*. Dan untuk nilai kV yang terdapat pada spesifikasi motor RC (*Radio Control*) yaitu menjelaskan berapa kali motor berputar dalam satu menit dengan tegangan yang berikan, tanpa motor tersebut diberi beban. Sebagai contoh apabila kita mempunyai 11.1V baterai dan nilai kV dari motor adalah 1600, maka motor dapat berputar $1600 \times 11.1 = 17760$ kali per menit. Nilai kV pada motor yang tinggi biasanya cocok untuk pesawat yang lebih kecil dengan baling-baling standar.

2.2. Sensor Rotary Encoder [1]



Gambar 2. 6 Rotary Encoder [1]

Rotary encoder adalah suatu komponen elektro mekanis yang memiliki fungsi untuk memonitoring posisi angular pada suatu poros yang berputar. Dari perputaran benda tersebut data yang termonitoring akan diubah ke dalam bentuk data digital oleh *rotary encoder* berupa lebar pulsa kemudian akan dihubungkan ke mikrokontroler.

Konstruksi *rotary encoder* berupa piringan tipis yang biasanya di kopel dengan poros yang berputar. Piringan tipis tersebut terdapat lubang di

sepanjang pinggir lingkarannya. Di bagian sisi-sisi piringan terdapat sebuah led dan *phototransistor* di bagian bersebrangan. Fungsi dari lubang-lubang yang berada di sepanjang pinggir lingkaran tersebut akan menghantarkan cahaya led ke *phototransistor*, sebaliknya jika cahaya led tidak menembus lubang piringan maka cahaya akan tertahan. Piringan tersebut akan berputar sesuai dengan kecepatan putaran motor sehingga *phototransistor* akan saturasi ketika cahaya LED menembus lubang-lubangnya.

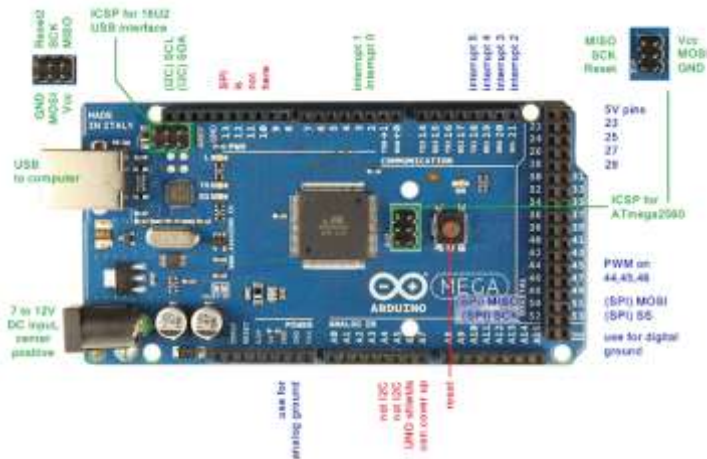
Pada saat saturasi *phototransistor* akan menghasilkan pulsa dengan *range* +0,5 V s/d +5 V.

2.3. Arduino Mega 2560

Board Arduino Mega 2560 adalah sebuah Board Arduino yang menggunakan ic Mikrokontroler ATmega 2560. Board ini memiliki Pin I/O yang relatif banyak, 54 digital Input / Output, 15 buah di antaranya dapat di gunakan sebagai output PWM, 16 buah analog Input, 4 UART. Arduino Mega 2560 di lengkapi kristal 16. Mhz Untuk penggunaan relatif sederhana tinggal menghubungkan power dari USB ke PC / Laptop atau melalui Jack DC pakai adaptor 7-12 V DC. Untuk lebih jelasnya dapat di lihat dari spesifikasi Arduino Mega 2560 di bawah ini :

Tabel 2. 2 Deskripsi Arduino Mega 2560

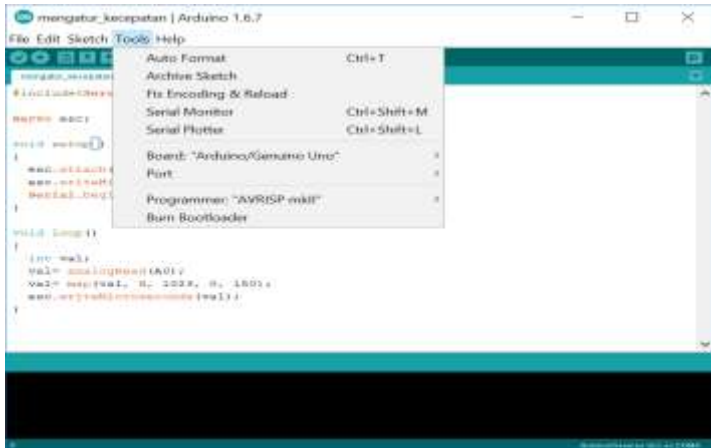
<i>Mikrokontroler</i>	<u>ATmega2560</u>
<i>Tegangan Operasional</i>	5V
<i>Tegangan Input (rekomendasi)</i>	7-12V
<i>Tegangan Input (limit)</i>	6-20V
<i>Pin Digital I/O</i>	54 (of which 15 provide PWM output)
<i>Pin Analog Input</i>	16
<i>Arus DC per Pin I/O</i>	20 mA
<i>Arus DC untuk Pin 3.3 V</i>	50 mA
<i>Memori Flash</i>	256 KB of which 8 KB used by bootloader
<i>SRAM</i>	8 KB
<i>EEPROM</i>	4 KB
<i>Clock Speed</i>	16 MHz
<i>LED_BUILTIN</i>	13
<i>Panjang</i>	101.52 mm
<i>Lebar</i>	53.3 mm
<i>Berat</i>	37 g



Gambar 2.7 Arduino Mega 2560

Untuk memprogram arduino juga harus dilakukan beberapa tahapan sebagai berikut :

1. *Setting Board Arduino*
 Dalam pemrograman *software* arduino harus di *setting* terlebih dahulu *board* arduino agar penggunaan arduino cocok. Dalam purwarupa kali ini arduino menggunakan arduino Mega 2560. Untuk *setting board* arduino bisa masuk ke *tools – board –* setelah itu pilihlah *board* arduino yang sesuai.
2. *Setting Serial*
Serial ini merupakan kabel arduino yang dihubungkan kepada komputer atau laptop. *Serial* ini mempunyai dua fungsi yang bisa digunakan. Pertama *serial port* digunakan untuk men-*download* program dari arduino yang kedua *serial* digunakan sebagai komunikasi *serial* pada arduino dengan komputer. *Setting serial* bisa masuk *tools – serial -* lali pilih COM yang sesuai dengan arduino yang terpasang. Untuk lebih jelasnya dapat dilihat pada Gambar 2.8.



Gambar 2. 8 Setting Serial Port

Apabila *program* tidak dapat di *download* karena *serial port*, maka cek terlebih dahulu *serial* yang benar pada *device manager*. Lalu dalam *software* arduino untuk memilih *serial port*nya samakan dengan *serial port* untuk arduino dalam *device manager* tersebut. Untuk masuk ke *device manager* dapat masuk start windows – lalu ketika *device manager* klik dua kali dan masuk ke dan COM.

2.3.1 Arduino Mega 2560 Pinout

Pin digital Arduino Mega2560 ada 54 Pin yang dapat di gunakan sebagai Input atau Output dan 16 Pin Analog berlabel A0 sampai A15 sebagai ADC, setiap Pin Analog memiliki resolusi sebesar 10 bit. Arduino Mega 2560 di lengkapi dengan pin dengan fungsi khusus, sebagai berikut :

- **Serial 4 buah** : Port Serial : Pin 0 (RX) dan Pin 1 (TX) ;Port Serial 1 : Pin 19 (RX) dan Pin 18 (TX); Port Serial 2 : Pin 17 (RX) dan Pin 16 (TX); Port Serial 3 : Pin 15 (RX) dan Pin 14 (TX). Pin Rx di gunakan untuk menerima data serial TTL dan Pin (Tx) untuk mengirim data serial TTL
- **External Interrupts 6 buah** : Pin 2 (Interrupt 0), Pin 3 (Interrupt 1), Pin 18 (Interrupt 5), Pin 19 (Interrupt 4), Pin 20 (Interrupt 3) dan Pin 21 (Interrupt 2)
- **PWM 15 buah** : 2,3,4,5,6,7,8,9,10,11,12,13 dan 44,45,46 pin-pin tersebut dapat di gunakan sebagai Output PWM 8 bit

- **SPI** : Pin 50 (MISO), Pin 51 (MOSI), Pin 52 (SCK), Pin 53 (SS) ,Di gunakan untuk komunikasi SPI menggunakan SPI Library
- **I2C** : Pin 20 (SDA) dan Pin 21 (SCL) , Komunikasi I2C menggunakan wire library
- **LED** : 13. Buit-in LED terhubung dengan Pin Digital 13

BAB III

PERANCANGAN DAN IMPLEMENTASI

Dalam bab ini akan membahas mengenai perancangan alat yang meliputi perencanaan perangkat keras (*hardware*) dan perangkat lunak (*software*).

Untuk memudahkan dalam pembahasan bab ini akan dibagi menjadi 3 yaitu:

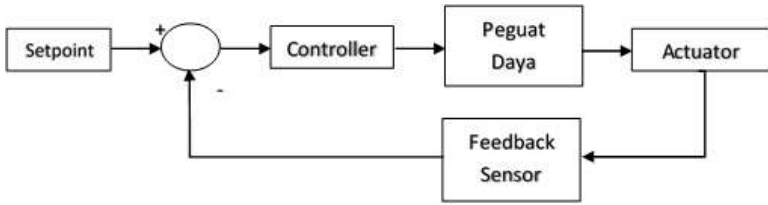
1. Blok fungsional sistem
2. Perancangan perangkat keras yang terdiri dari perancangan mekanik dan elektrik, yaitu:
 - a. Perancangan mekanik meliputi desain alat secara keseluruhan
 - b. Perancangan elektrik meliputi rangkaian sensor *rotary encoder*
3. Perancangan perangkat lunak yang terdiri dari :
 - a. Perancangan perangkat lunak dengan *software* Arduino Mega 2560 yang menggunakan bahasa C sebagai bahasa pemrogramannya.

3.1. Blok Fungsional Sistem

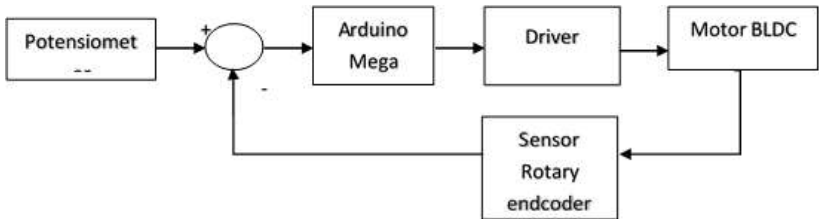
Perancangan sistem dalam pembuatan alat ini secara garis besar diserti urutan dan cara kerja alat ini di ilustrasikan pada Gambar 3.1 dan Gambar 3.2.

Dari Gambar 3.1 dapat dilihat bahwa sistem tersebut terdiri dari beberapa blok fungsional yaitu;

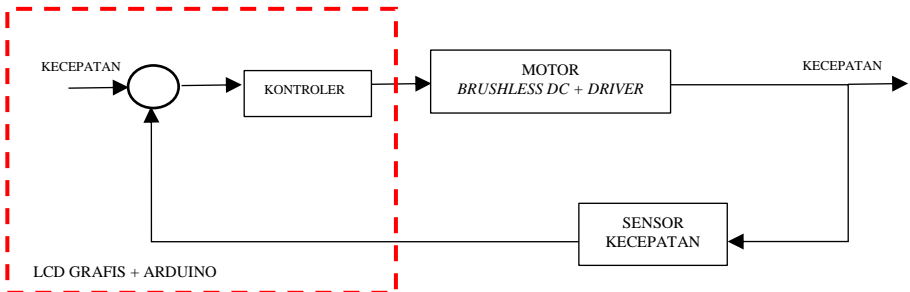
1. Arduino Mega 2560, merupakan mikrokontroler yang berfungsi sebagai *interface* dari perangkat elektronik dan dapat menyimpan program didalamnya.
2. *Driver* motor yaitu ESC berfungsi sebagai penggerak motor dan sensor.
3. Motor *brushless* DC, digunakan sebagai objek yang akan dikontrol kecepatannya.
4. Sensor kecepatan (*rotary encoder*), digunakan untuk mengetahui sampai seberapa kecepatan motor yang terjadi.
5. Sensor arus digunakan untuk mengetahui nilai arus pada saat motor berputar.
6. Tampilan (*Display*), untuk tampilan digunakan GLCD 128x64, GLCD 128x64 ini digunakan untuk menampilkan kecepatan motor *brushless* DC.



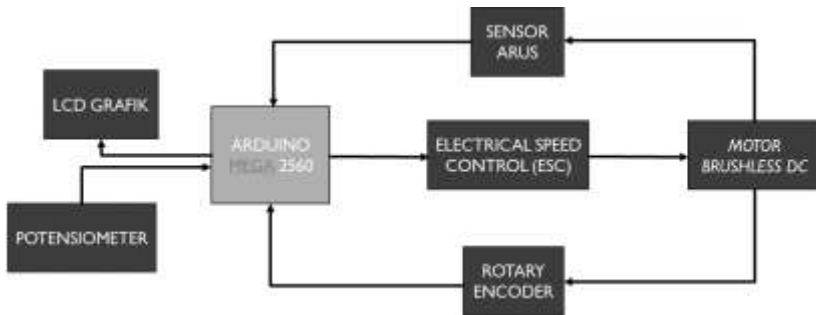
Gambar 3. 2 Konsep Blok Diagram Sistem Kontrol Kecepatan



Gambar 3. 3 Blok Diagram Sistem Kontrol Kecepatan



Gambar 3. 4 Blok Diagram Sistem Close Loop



Gambar 3. 1 Diagram Blok Alur Sistem

3.2. Perancangan Perangkat Keras (*Hardware*)

Pada perancangan perangkat keras ini, prosesnya dibagi menjadi 2 bagian, yaitu perancangan mekanik dan elektrik. Masing-masing perancangan tersebut selanjutnya akan dibahas lebih mendalam pada sub bab berikutnya.

3.2.1 Perancangan Mekanik

Perancangan mekanik ini terdiri dari desain alat secara keseluruhan, motor *brushless* DC, *Rotary Encoder*, *Flywheel* dan *driver* motor *brushless* DC

3.2.1.1 Motor Brushless DC

Motor BLDC dipilih karena memiliki berbagai kelebihan, yaitu, dapat menggerakkan dengan daya yang besar, suaranya halus, torsi besar, efisiensi tinggi, mudah di control, dan perawatan yang rendah. Motor BLDC yang digunakan merupakan motor BLDC yang biasa digunakan untuk *aero-modelling* sehingga ukurannya lebih kecil.. Untuk lebih jelasnya mengenai motor ini dapat dilihat pada Gambar 2.1 dan Tabel 2.1.

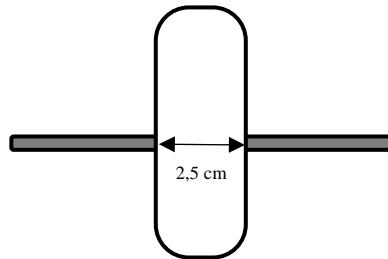
3.2.1.2 Electronic Speed Control (ESC)

ESC atau disebut juga *Electronic Speed Control* adalah *driver* penggerak untuk jenis motor *brushless*, biasanya digunakan pada bidang *aeronautical* atau RC (*Radio Control*). *Driver* ini bertujuan untuk mengubah sinyal kontrol yang dikirimkan Arduino menjadi tegangan 3 fasa beda 120 derajat yang digunakan untuk menggerakkan motor BLDC. Untuk menentukan ESC yang akan kita gunakan sangatlah penting untuk mengetahui kekuatan (*peak*

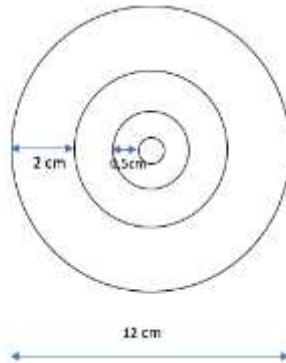
current) dari motor. Pilihlah ESC yang kekuatannya melebihi kekuatan motor.

3.2.1.3 Flywheel

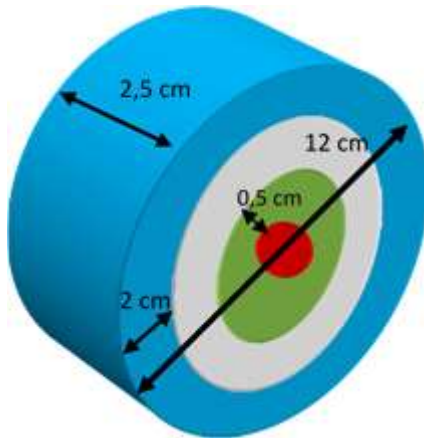
Flywheel (roda gila) merupakan perangkat untuk menyimpan energi rotasi, perangkat ini mendapatkan sumber energi dari percepatan motor BLDC. Dan kemudian perangkat ini meneruskan energi rotasi ke motor generator yang membuat motor generator menghasilkan tegangan. Flywheel yang kami gunakan ini berbahan acrylic pada bagian beban dan penyangganya.



Gambar 3.3 beban flywheel tampak samping.



Gambar 3.5 beban flywheel tampak depan



Gambar 3.6 Ilustrasi bentuk *Flywheel*

Acrylic yang telah di cutting, disusun dan di rangkai menyerupai flywheel. Gambar 3.4 tebal roda beban yang berasal dari 3 lapis cutting acrylic. Lapisan pertama berbentuk donat dengan diameter 12 cm, 2 cm ketebalan dri sisi luar dan 0,5 cm ketebalan bagian dalam seperti pada gambar 3.5. Lapisan kedua berbentuk jaring-jaring roda. Lapis pertama dan ketiga memiliki bentuk yang sama

3.2.2 Perancangan Elektrik

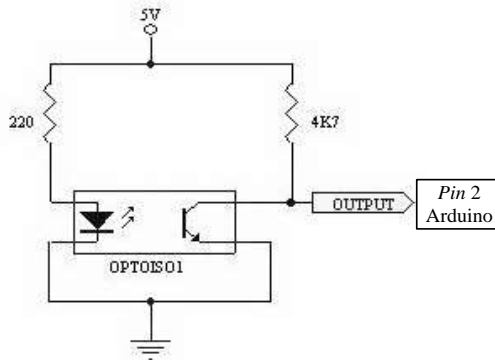
Perancangan elektronik ini meliputi desain skematik layout PCB serta pengkabelan. Rangkaian elektrik pada *plant* ini meliputi rangkaian sensor *rotary encoder*, dan Arduino Mega 2560 .

3.2.2.1. Rangkaian Sensor *Rotary Encoder*

Rotary encoder pada Proyek Akhir ini akan digunakan untuk menentukan banyaknya putaran poros tiap menit (RPM) yang kemudian akan menghasilkan gelombang kotak yang frekuensinya akan bertambah bila kecepatan putar poros bertambah. *Rotary encoder* ini diletakkan pada poros yang sudah dikopel dengan motor *brushless DC*.

Rotary encoder ini tersusun dari suatu piringan tipis yang memiliki lubang-lubang yang terdapat pada piringan tersebut. Setelah itu akan ditempatkan LED pada salah satu sisi piringan. Hal ini akan membuat cahaya masuk menuju piringan.

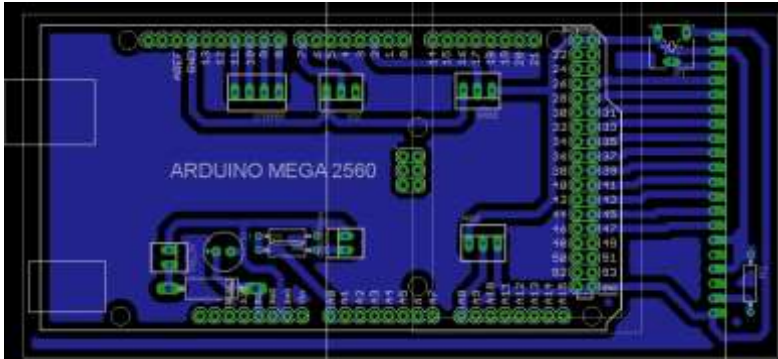
Kemudian disisi lain dari piringan ini diletakan *phototransistor* yang bertujuan untuk mendeteksi cahaya LED yang berseberangan. Piringan tipis ini yang nantinya akan dikopel dengan poros motor ataupun perangkat berputar lainnya yang ingin kita ketahui posisinya, hal ini akan membuat piringan berputar ketika motor tersebut berputar. Apabila cahaya yang berasal dari LED mencapai *phototransistor*, maka *phototransistor* itu akan mengalami saturasi dan akan menghasilkan suatu pulsa gelombang persegi.



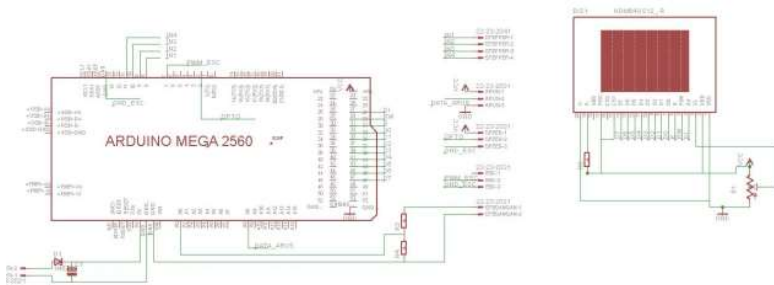
Gambar 3. 1 Rangkaian Rotary Encoder

3.2.2.2. Rangkaian Arduino Mega 2560

Rangkaian Arduino Mega pada Proyek Akhir ini digunakan sebagai core dari segala komponen yang menghubungkan satu komponen dengan komponen lain secara terstruktur. Sehingga setiap sistem dapat bekerja dengan sesuai fungsi dari komponen. Untuk itu dibutuhkan rangkaian yang tepat sehingga menghasilkan nilai yang diinginkan. Hal ini dapat dilihat dari Gambar 3.5 dan Gambar 3.6.



Gambar 3.5 Layout Arduino Mega 2560



Gambar 3.6 Skematik Arduino Mega 2560 dan Komponen

Dari Gambar 3.5 merupakan layout yang menunjukkan pinout dari Arduino Mega. Sedangkan Gambar 3.6 merupakan skematik Arduino dengan beberapa komponen lain, seperti Sensor arus, sensor tegangan, motor stepper, motor brushless dan GLCD. Masing – masing komponen memiliki fungsi sendiri yang saling berkaitan, sehingga dibutuhkannya penghubung untuk menyinkronkan semua komponen tersebut.

3.3. Perancangan Perangkat Lunak (*Software*)

Agar kecepatan motor brushless dan motor stepper dapat dikontrol maka perlu drancang secara software yang mampu mengelola dan mengontrol data pada setiap komponen. Untuk motor brushless dapat dikontrol kecepatan dan hasil putaran tersebut dapat dilihat di LCD. Sedangkan untuk

motor stepper dapat mendorong generator sehingga terjadi gesekan dengan *flywheel*.maka perlu di rancang sebuah *software* yang mampu mengelola dan mengontrol data terhadap kinerja peralatan pengereman regeneratif. Software merupakan program berisi perintah-perintah yang dieksekusi oleh arduino mega 2560 sehingga sistem dapat bekerja sesuai dengan alur dan tujuan yang dirancang.

3.3.1. Pemrograman *Software* Arduino

Dalam perancangan program pada *software* arduino dengan fungsi terkait yang dibutuhkan diperlukan beberapa tahapan yang harus dilakukan terlebih dahulu. Tahapan tersebut adalah membuat algoritma dari alat yang sudah kita jalankan.

Pembuatan algoritma ini dilakukan setelah membuat *flowchart*, dari algoritma kita ini maka diharapkan alat yang akan dibuat ini dapat terlebih lebih sederhana. Setelah tahapan tersebut terselesaikan barulah kita memprogram fungsi terkait yang dikodingkan dalam bahasa C.

Berikut ini algoritma program utama dari kontrol kecepatan :

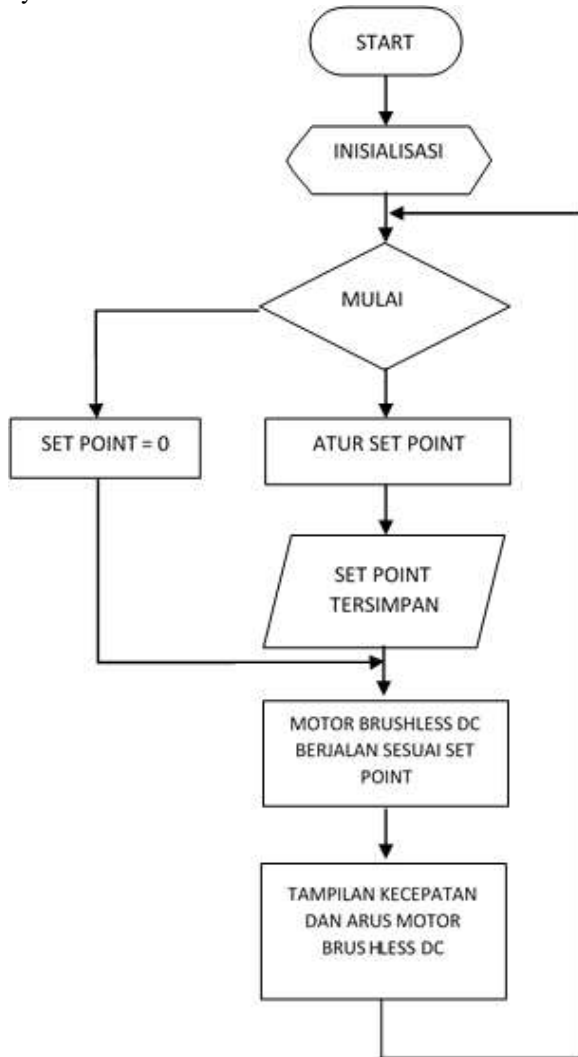
1. Modul dapat di operasikan setelah sistem terpasang dengan benar seperti motor *brushless* DC, *driver* motor *brushless* DC, catu daya dan *potensiometer*.
2. Modul dapat bekerja jika rangkaian kontrol sudah dijalankan dan sudah terpasang.
3. *Potensiometer* digunakan untuk mengatur *range* yang akan di suplai ke *driver* yang berupa tegangan dan akan di ubah menjadi digital melalui ADC Arduino.

Pada Proyek Akhir ini *software* yang digunakan adalah program Arduino Mega 2560 untuk membuat dan merencanakan program dalam bahasa C. Pemrograman *software* arduino dirancang dengan menggunakan *software* yang bernama Arduino IDE dengan menggunakan bahasa pemrograman C.

Arduino sangatlah berbeda sekali dengan mikrokontroler. Arduino merupakan sebuah *kit* mikrokontroler AVR yang dibuat dalam sebuah **board** (papan PCB). Dikembangkan di Italia sejak tahun 2005. Dalam 1 *board* sudah terdapat mikrokontroler lengkap dengan *pin/port* untuk koneksi serta sudah dilengkapi dengan *downloader*. Dalam segi bahasa pemrograman, arduino memiliki bahasa pemrograman yang lebih mudah dan sederhana terutama bagi pemula.

Alasan bahasa pemrograman arduino lebih mudah dan sederhana adalah karena didalam arduino sudah terdapat beberapa *library* yang dapat digunakan untuk merancang pemrograman yang diinginkan. Pada Gambar

3.5 berikut menunjukkan *flowchart* yang digunakan pada Arduino Mega 2560 pada Proyek Akhir ini.



Gambar 3.7 *Flowchart* Arduino Mega 2560

-----Halaman ini sengaja dikosongkan-----

BAB IV

PENGUJIAN DAN ANALISA ALAT

Pada bab ini akan dibahas mengenai pengujian alat dan analisa data dari hasil rancangan alat yang telah dibuat. Pengujian alat ini ditujukan untuk memastikan agar peralatan dapat berfungsi dengan baik.

Pada gambar 4.1 Pengujian dan analisa dari alat ini meliputi pengujian sensor *rotary encoder*, pengujian kecepatan putaran motor dan arus *brushless* DC tanpa beban, serta pengujian kecepatan putaran motor dan arus *brushless* DC dengan beban. Setelah melakukan beberapa pengujian tersebut, data yang diperoleh akan dianalisa untuk mengetahui proses kerja dari seluruh sistem alat yang dibuat.



Gambar 4.1 Realisasi Plant Motor BLDC

4.1. Pengujian Sensor *Rotary Encoder*

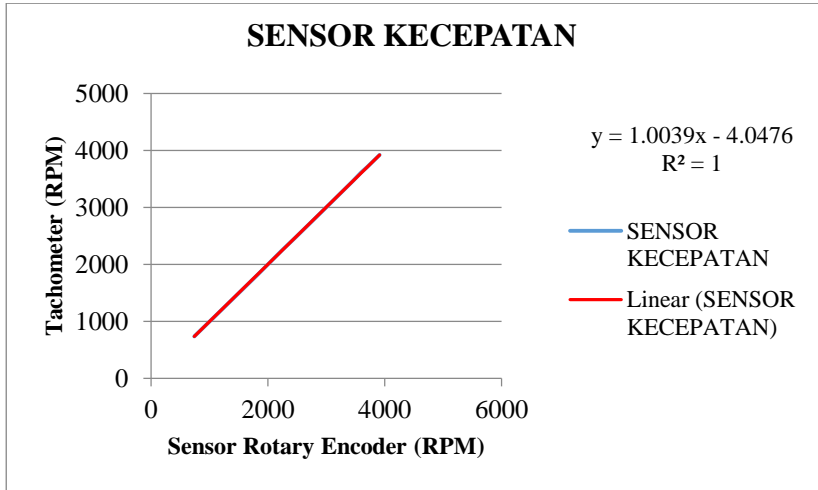
Tabel 4.1 Kalibrasi Sensor *Rotary Encoder*

No	Kecepatan Motor BLDC (tachometer) RPM	Kecepatan Motor BLDC (Encoder) RPM	Kalibrasi (RPM)	Error
1	737.5 RPM	737.8 RPM	736.3287 RPM	0.20%
2	1817 RPM	1816.4 RPM	1820.039 RPM	0.20%
3	2255 RPM	2245 RPM	2259.747 RPM	0.66%
4	2472 RPM	2470 RPM	2477.593 RPM	0.31%
5	2978 RPM	2973 RPM	2985.567 RPM	0.42%
6	3468 RPM	3458 RPM	3477.478 RPM	0.56%
7	3610 RPM	3595 RPM	3620.031 RPM	0.70%
8	3713 RPM	3705 RPM	3723.433 RPM	0.50%
9	3917 RPM	3906 RPM	3928.229 RPM	0.57%
10	3922 RPM	3911 RPM	3933.248 RPM	0.57%

Tabel 4.1 menunjukkan hasil dari pengujian kalibrasi kecepatan motor *brushless* DC. Penjelasan data pada Tabel 4.1 yaitu dengan membandingkan korelasi dari pengukuran menggunakan sensor *rotary encoder* dan tachometer.

Pada pengujian yang tertera pada Tabel 4.1 dapat dilihat bahwa *output* hasil pembacaan pada sensor *rotary encoder* dan *digital tachometer* memiliki perbedaan hingga mencapai kurang lebih 10-20 RPM. Perbedaan kecepatan ini semakin besar pada saat kecepatan tinggi. Hal ini disebabkan karena getaran dari poros motor yang sudah tidak lurus lagi yang menyebabkan getaran piringan *encoder* saat diputar pada kecepatan tinggi.

Adanya keterangan *error* dikarenakan pembacaan pada sensor tidak terlalu akurat seperti pada alat ukur sebenarnya.



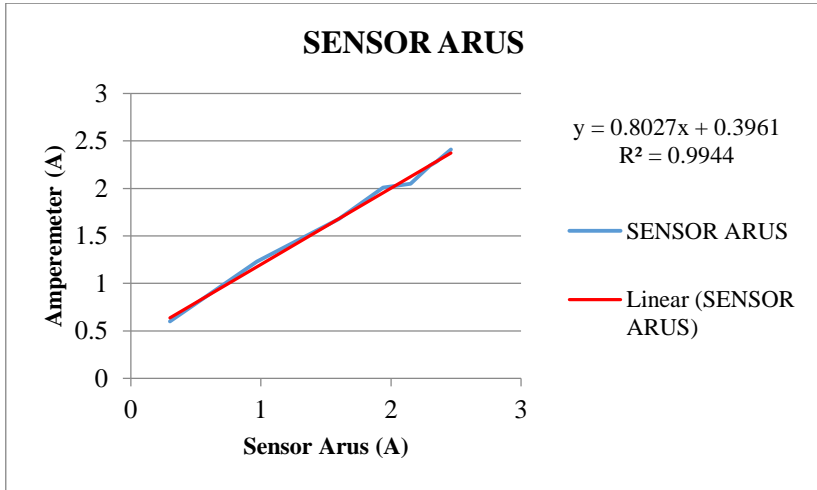
Gambar 4.2 Grafik Linier Kalibrasi Sensor Kecepatan

Untuk mendapatkan data yang sesuai dan benar maka diperlukan proses kalibrasi. Pada proses kalibrasi ini membutuhkan 2 data kecepatan yang berbeda yaitu didapatkan dari pembacaan dari kecepatan motor dari rotary encoder dan data pembacaan kecepatan menggunakan Tachometer. Data tersebut dapat dilihat pada tabel 4.1 untuk mengetahui grafik pembacaan sensor tegangan dapat dilihat pada gambar 4.2. setelah didapatkan data maka dari 2 data tersebut dimasukkan kedalam rumus sebagai berikut :

Kalibrasi = (Kecepatan Motor BLCD (Encoder) * 1.0039) - 4.0476

Setelah didapatkan data kalibrasi maka bisa dicari nilai *Error* suatu pembacaan sensor tersebut dengan dimasukkan kedalam rumus sebagai berikut :

Error :
$$\frac{(\text{Nilai Kalibrasi} - \text{Data kecepatan Tachometer})}{\text{Data Kecepatan Tachometer}}$$



Gambar 4.3 Grafik Linier Kalibrasi Sensor Arus

Pada Gambar 4.3 merupakan Grafik linier sensor arus yang didapatkan, sehingga dari grafik tersebut di dapat nilai kalibrasi dan nilai error pembacaan sensor. Pada proses kalibrasi ini membutuhkan 2 data arus yang berbeda yaitu didapatkan dari pembacaan dari arus motor dari sensor arus dan data pembacaan kecepatan menggunakan Amperemeter. Data tersebut dapat dilihat pada tabel 4.1 untuk mengetahui grafik pembacaan sensor arus dapat dilihat pada gambar 4.3.

4.2 Pengujian Kecepatan Putaran Motor *Brushless* DC Tanpa Beban Pengereman

Untuk melakukan pengujian terhadap data tabel dibawah ini yaitu menghidupkan Arduino Mega dengan menekan saklar On, kemudian *ESC* akan aktif berjalan, setelah itu masukkan program ke Arduino Mega untuk mengatur *Duty Cycle* pada potensiometer motor, setelah itu jalankan motor *BLDC* lewat potensiometer motor yang sudah di *setting* pada program, nanti hasil akan keluar pada LCD yang sudah di program berdasarkan apa yang telah di baca pada sensor *rotary encoder*, maka mengambil data kecepatan pada motor dilakukan lewat hasil pembacaan sensor *Rotary Encoder*, selain menampilkan kecepatan alat akan membaca data nilai Arus pada motor *Brushless* DC yang akan ditampilkan pada LCD dengan bentuk Grafik dan

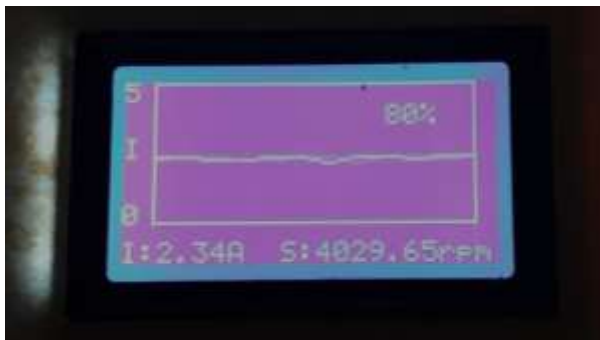
nilai Amper. Untuk mengambil data *Ampere* pada motor dilakukan lewat Sensor Arus. Hasil pengujian Sensor kecepatan dan Sensor Arus dapat dilihat pada Tabel 4.2.

Tabel 4.2 Pengujian Kecepatan dan Arus Motor BLDC dan Arus Tanpa Beban

No	Input Potensio Motor (<i>Duty Cycle</i> %)	Kecepatan Motor BLDC (Encoder) RPM	Arus Motor (Alat)	Tegangan Motor BLDC (Volt)
1	10 %	934,50 RPM	0,83 A	0,9 Volt
2	20 %	2096,63 RPM	1,24 A	2,1 Volt
3	30 %	2803,15 RPM	1,69 A	3,2 Volt
4	40 %	3278,56 RPM	1,94 A	4,6 Volt
5	50 %	3592,20 RPM	2,00 A	5,8 Volt
6	60 %	3785,34 RPM	2,22 A	6,9 Volt
7	70%	3924,00 RPM	2,32 A	8,1 Volt
8	80%	4029,65 RPM	2,34 A	9,4 Volt
9	90%	4239,30 RPM	2,36 A	10,6 Volt
10	100%	4239,30 RPM	2,40 A	11,42 Volt



Gambar 4.4 Pengujian Kecepatan Putaran Motor dan Arus Motor *Brushless* DC tanpa beban dengan nilai *Duty Cycle* 10%



Gambar 4.5 Pengujian Kecepatan Putaran Motor dan Arus Motor *Brushless* DC tanpa beban dengan nilai *Duty Cycle* 80%

Pada gambar 4.4 dan gambar 4.5 merupakan hasil dari pembacaan sensor kecepatan motor dan sensor arus yang terbaca oleh Arduino dan ditampilkan pada LCD dengan bentuk grafik linier arus dan nilai tertulis.

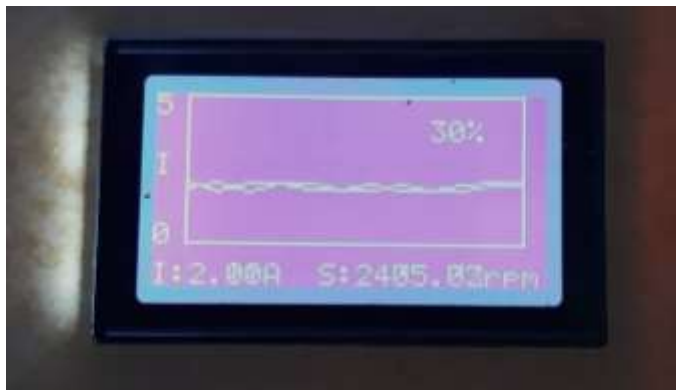
4.3 Pengujian Kecepatan Putaran Motor *Brushless* DC dengan Beban Pengereman

Untuk melakukan pengujian terhadap data tabel dibawah ini yaitu menghidupkan Arduino Mega dengan menekan saklar On, kemudian *ESC* akan aktif berjalan, setelah itu masukkan program ke Arduino Mega untuk mengatur *Duty Cycle* pada potensiometer motor, setelah itu jalankan motor *BLDC* lewat potensiometer motor yang sudah di *setting* pada program, setelah motor berputar sesuai nilai input *duty cycle* kemudian motor generator beban pengereman diaktifkan, dengan cara menggeser motor sampai mengenai tuas pengereman dan motor dikunci agar tidak berubah ubah hasilnya. nanti hasil akan keluar pada LCD yang sudah di program berdasarkan apa yang telah di baca pada sensor *rotary encoder*, maka mengambil data kecepatan pada motor dilakukan lewat hasil pembacaan sensor *Rotary Encoder*, selain menampilkan kecepatan alat akan membaca data nilai Arus pada motor *Brushless* DC yang akan ditampilkan pada LCD dengan bentuk Grafik dan nilai Amper. Untuk mengambil data *Ampere* pada motor dilakukan lewat Sensor Arus. Hasil pengujian Sensor kecepatan dan Sensor Arus dapat dilihat pada Tabel 4.3.

Tabel 4.3 Pengujian Kecepatan dan Arus Motor BLDC dengan Beban Motor Generator

9	<i>Input Potensio Motor (Duty Cycle %)</i>	Kecepatan Motor BLDC (Encoder) RPM	Arus Motor (Alat)	Tegangan Motor BLDC (Volt)
1	10 %	596,10 RPM	0,74 A	0,7 Volt
2	20 %	1667,43 RPM	1,43 A	1,9 Volt
3	30 %	2405,02 RPM	2,00 A	3,0 Volt
4	40 %	2933,56 RPM	2,36 A	4,4 Volt
5	50 %	3265,36 RPM	2,68 A	5,3 Volt
6	60 %	3512,97 RPM	2,93 A	6,7 Volt

No	Input Potensio Motor (Duty Cycle %)	Kecepatan Motor BLDC (Encoder) RPM	Arus Motor (Alat)	Tegangan Motor BLDC (Volt)
7	70%	3640,08 RPM	3,15 A	7,9 Volt
8	80%	3843,12 RPM	2,96 A	9,1 Volt
9	90%	4188,12 RPM	2,78 A	10,3 Volt
10	100%	4117,14 RPM	3,36 A	11,2 Volt



Gambar 4.6 Pengujian Kecepatan Putaran Motor dan Arus Motor *Brushless* DC dengan beban dengan nilai *Duty Cycle* 30%



Gambar 4.7 Pengujian Kecepatan Putaran Motor dan Arus Motor *Brushless* DC dengan beban dengan nilai *Duty Cycle* 70%

Pada gambar 4.6 dan gambar 4.7 merupakan hasil dari pembacaan sensor kecepatan motor dan sensor arus yang terbaca oleh Arduino dan ditampilkan pada LCD dengan bentuk grafik linier arus dan nilai tertulis.

Setelah mendapatkan hasil pengujian kecepatan motor brushless DC dengan beban pengereman dan tanpa beban pengereman ini, untuk kecepatan motor dengan beban ini terdapat perlambatan kecepatan jika dibandingkan dengan tanpa beban. Untuk perlambatan yang dihasilkannya pun tidak terlalu terlihat, tetapi jika di analisis secara mendalam maka terdapat perbedaan kecepatan sekitar 100-300 RPM disetiap kenaikan nilai input potensiometer (*Duty Cycle* %).

Sedangkan untuk hasil pengujian arus pada motor brushless DC dengan beban dan tanpa beban ini, perbedaan ini sangat kecil, jika dilihat dari tabel 4.2 dan tabel 4.3 nilai arus yang terbaca pada sensor arus tersebut berbeda, hanya selisih 0,2 – 0,9 Ampere,

Jadi kecepatan motor brushless DC tanpa beban dengan beban ini terdapat penurunan, hal ini dibuktikan dengan adanya perlambatan kecepatan. tetapi untuk nilai arus terdapat kenaikan tetapi tidak terlalu besar.

-----Halaman ini sengaja dikosongkan-----

BAB V

PENUTUP

Bab penutup ini berisi tentang kesimpulan yang diperoleh selama proses pembuatan modul kontrol kecepatan motor *brushless* DC, kesimpulan dari hasil pengujian dan analisa data, serta saran untuk modul kontrol kecepatan motor *brushless* DC ini kedepannya.

5.1. Kesimpulan

1. Pada perancangan kontrol kecepatan motor *brushless* DC sebagai modul pembelajaran ini dapat mengukur nilai kecepatan motor brushless DC (RPM) dengan nilai error 0,2 % - 0,57 %.
2. Pada perancangan kontrol kecepatan motor *brushless* DC sebagai modul pembelajaran ini dapat mengukur nilai Arus pada motor Brushless DC tanpa beban dan menggunakan beban dengan nilai selisih 0,2 Ampere sampai dengan 0,9 Ampere.
3. Pada perancangan kontrol kecepatan motor *brushless* DC sebagai modul pembelajaran ini dapat mengukur nilai kecepatan motor brushless DC (RPM) menggunakan beban dan tanpa beban dengan selisih nilai perlambatan 100 RPM - 300 RPM.

5.2 Saran

1. Diharapkan kedepannya dapat menggunakan Driver motor atau ESC sendiri.
2. Diharapkan menggunakan Motor Brushless RPM dan dimensi yang lebih besar.
3. Diharapkannya menggunakan motor generator yang lebih besar daya dan dimensinya.
4. Keseluruhan alat baik penyangga maupun tatanan lebih baik dibuat dari bahan yang kuat dan tahan terhadap getaran, agar sewaktu dijalankan tidak mengganggu kecepatan putar.

-----Halaman ini sengaja dikosongkan-----

DAFTAR PUSTAKA

- [1] Hudaibiy Hibban. “*Desain Kontroler Fuzzy PID Gain Schedulling Untuk Pengaturan Kecepatan Motor DC Tanpa Sikat*”, **Proyek Akhir**, Jurusan Teknik Elektro FTI-ITS, Surabaya, 2015.
- [2] Baharsyah, Baharuddin. “*Pengaturan Kecepatan Motor Brushless Dc Menggunakan Kontroler Berbasis PID*”, **Proyek Akhir**, Jurusan Departemen Teknik Eletro Otomasi FV-ITS, Surabaya, 2019
- [3] Bachtiar, Muhammad Fachri, Priyatna, Alif Gigah. **Perancangan Rem Magnetik Pada Motor DC Dengan Menggunakan Arduino**, Proyek Akhir, D3 Teknik Elektro, ITS Surabaya, 2015.
- [4] Nur Robi Annisa dan Zaka Perwira, **Pembuatan Modul Kontrol Kecepatan Motor Brushless DC dengan Mikrokontroler**, **Proyek Akhir**, Program D3 Teknik Elektro FTI-ITS, Surabaya, 2016.

-----Halaman ini sengaja dikosongkan-----

LAMPIRAN A ***LISTING PROGRAM***

LISTING PROGRAM ARDUINO MEGA 2560

```
#include "ESC.h"
#include "TimerThree.h"
#include <glcd.h>
#include "fonts/Arial14.h" // proportional font
#include "fonts/SystemFont5x7.h" // system font

#if DISPLAY_HEIGHT < 64
#error ks0108 example requires a display at least 64 pixels tall
#endif
#if DISPLAY_WIDTH < 128
#error ks0108 example requires a display at least 128 pixels wide
#endif

int data[12];
float arus;

float kec ;
float gArus;

#define POT_PIN (A2)
#define SKC (21)
#define SAR (A8)

double pulsa=0;
float kecepatan;
int x;
int duty;
```

```

float mvperamp = 185.0;
int rawval = 0;
float acsoff = 2500.0;
float voltage=0;
float amps =0;
float lastval=0;
void isr(){
    kecepatan=(float)pulsa*60.0/36.0;
    if(kecepatan>0){
        kecepatan =(0.9866*kecepatan) + 5.8555;
        kecepatan = (kecepatan*1.0039) - 4.0476;}
    pulsa=0;
}

ESC myESC (7, 1000, 2000, 500);
int val;
void setup() {
    Timer3.initialize(1000000);
    Timer3.attachInterrupt(isr);
    myESC.arm();
    pinMode(SKC , INPUT_PULLUP);
    GLCD.Init(NON_INVERTED);
    GLCD.ClearScreen();

    GLCD.SelectFont(System5x7);

    GLCD.ClearScreen();
    GLCD.DrawRect(10, 0, 110, 49, BLACK);
    GLCD.GotoXY(0,0);          // positon cursor
    GLCD.Puts("5");           // print a text string
    GLCD.GotoXY(0,43);        // positon cursor
    GLCD.Puts("0");           // print a text string

    attachInterrupt(2,rutinInterupsi0,RISING);

    Serial.begin(9600);
}
void loop(){

```

```

for(int i=0;i<20;i++){
  rawval=analogRead(SAR);
  voltage = (rawval/1023.0)*5000.0;
  amps=((voltage-acsoff)/mvperamp);
  lastval=lastval+amps;
}
amps=(lastval/-20.0)+0.123;
amps=(amps*1.0342)-0.3612;
amps = (amps*0.8027)+0.3961;
if(amps<0.3){amps=0.02;}
lastval=0;
Serial.println(amps,3);

  bldc_kontrol();

  tampil_grafik_arus();

  delay(100);
}
void bldc_kontrol(){
  val = analogRead(POT_PIN);
  val = map(val, 0, 1023, 1000, 2000);
  myESC.speed(val);
  duty=map(val,1000,2000,0,100);
}

void tampil_grafik_arus(){
  for(int i=0;i<12;i++){
    gArus=50-(amps*10.0);
    if(i==11){data[i]=(int)gArus;}

    else {data[i]=data[i+1];}
  }
  arus=5-(data[11]/10.0); //
  GLCD.ClearScreen();
  GLCD.DrawRect(10, 0, 110, 49, BLACK);
  GLCD.GotoXY(0,0);
  GLCD.Puts("5");
  GLCD.GotoXY(0,20);

```

```

GLCD.Puts("I");
GLCD.GotoXY(0,43);
GLCD.Puts("0");
GLCD.CursorTo(15,1);
GLCD.Puts((String)duty);
GLCD.Puts("% ");
GLCD.CursorTo(0,7);
GLCD.Puts("I:");
GLCD.Puts((String)amps);
GLCD.Puts("A");
GLCD.CursorTo(9,7);
GLCD.Puts("S:");
GLCD.Puts((String)kecepatan);
GLCD.Puts("rpm");
GLCD.DrawLine( 10, data[0], 20, data[1], BLACK);
GLCD.DrawLine( 20, data[1], 30, data[2], BLACK);
GLCD.DrawLine( 30, data[2], 40, data[3], BLACK);
GLCD.DrawLine( 40, data[3], 50, data[4], BLACK);
GLCD.DrawLine( 50, data[4], 60, data[5], BLACK);
GLCD.DrawLine( 60, data[5], 70, data[6], BLACK);
GLCD.DrawLine( 70, data[6], 80, data[7], BLACK);
GLCD.DrawLine( 80, data[7], 90, data[8], BLACK);
GLCD.DrawLine( 90, data[8], 100, data[9], BLACK);
GLCD.DrawLine( 100, data[9], 110, data[10], BLACK);
GLCD.DrawLine( 110, data[10], 120, data[11], BLACK);

}
void rutinInterupsi0(){ pulsa++;}

```

LAMPIRAN B

DATASHEET

1. Datasheet Arduino Mega 2560



Arduino Mega 2560 Datasheet



Overview

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 ([datasheet](#)). It has 54 digital input/output pins (of which 16 can be used as PWM outputs), 16 analog inputs, a UART (hardware serial port), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino [Duemilanove](#) or [Diecimila](#).

Schematic & Reference Design

Eagle file: [atmega2560-reference-design.sch](#)

Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

Power

The Arduino Mega can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1 mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the **GND** and **V_{CC}** pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the ATmega8U2 programmed as a USB-to-serial converter.

The power pins are as follows:

- **V_{IN}**. The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V**. The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3**. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND**. Ground pins.

Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX)**. Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2)**. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM: 0 to 13**. Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- **SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS)**. These pins support SPI communication using the [SPI library](#). The SPI pins are also broken out on the ICSP header, which is physically compatible with the Uno, [Duemilanove](#), and [Diecimila](#).
- **LED: 13**. There is a built-in LED connected to digital pin 13. When the pin is HIGH



[value](#), the LED is on, when the pin is LOW, it's off.

- **I²C: 20 (SDA) and 21 (SCL)**. Support I²C (TWI) communication using the [Wire library](#) (documentation on the Wiring website). Note that these pins are not in the same location as the I²C pins on the [Duemilanove](#) or [Diecimila](#).

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and [analogReference\(\)](#) function.

There are a couple of other pins on the board:

- **AREF**. Reference voltage for the analog inputs. Used with [analogReference\(\)](#).
- **Reset**. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

Communication

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically). The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Mega2560's digital pins.

The ATmega2560 also supports I²C (TWI) and SPI communication. The Arduino software includes a [Wire](#) library to simplify use of the I²C bus; see the [documentation on the Wiring website](#) for details. For SPI communication, use the [SPI library](#).

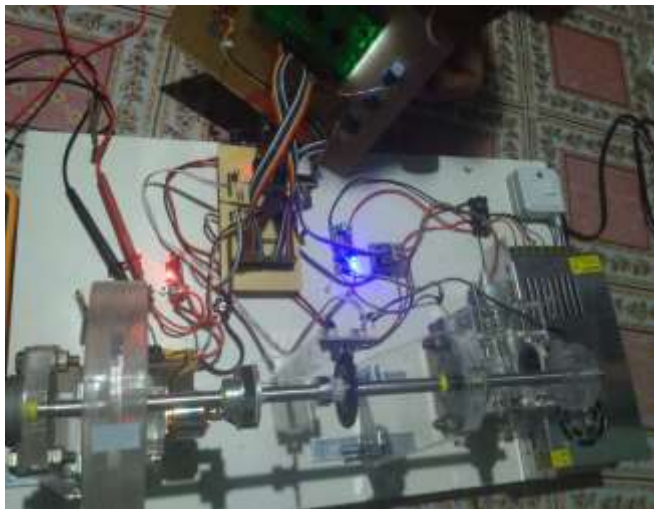
Programming

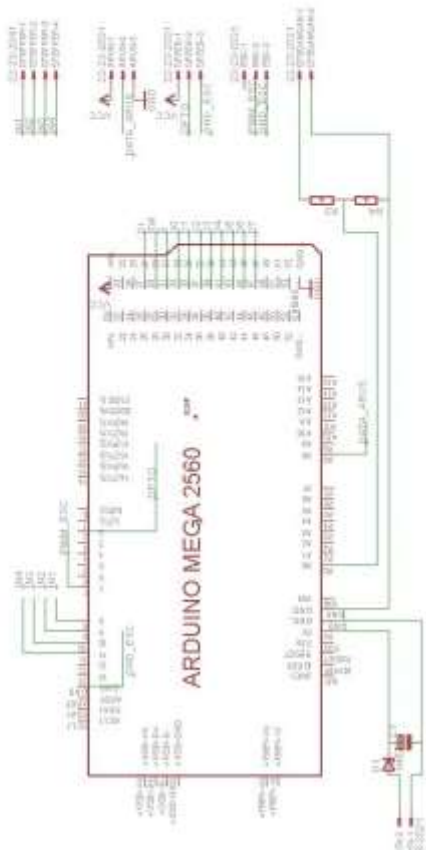
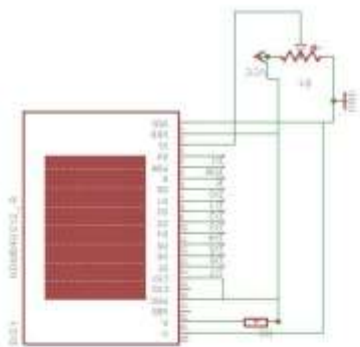
The Arduino Mega can be programmed with the Arduino software ([download](#)). For details, see the [reference](#) and [tutorials](#).

The ATmega2560 on the Arduino Mega comes [preburned](#) with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It

-- *Halaman ini sengaja dikosongkan* --

LAMPIRAN C
DOKUMENTASI





RIWAYAT PENULIS



Nama : Muchamad Rizki Abdillah
TTL : Gresik, 18 Mei 1997
Jenis Kelamin : Laki - Laki
Agama : Islam
Alamat Asal : Jl. Sunan Giri No. 79
Kawisanyar Kebomas-
Gresik
Telp/HP : 085852310830
E-mail : rizkiabd79@gmail.com

RIWAYAT PENDIDIKAN

- 2003 – 2009 : *MI Modern Sunan Giri (YIS)*
- 2009 – 2012 : *MTS Ma'arif Sidomukti*
- 2012 – 2015 : *SMAN 1 Kebomas Gresik*
- 2015 – Sekarang : *Departemen Teknik Elektro Otomasi Institut Teknologi Sepuluh Nopember.*

PENGALAMAN KERJA

- *Kerja Praktek PT SEMEN INDONESIA TUBAN (Juni 2017 – Juli 2017)*
- *Kerja Praktek PT PLN (Persero) Distribusi Jawa Timur Rayon (Juli 2017 – Agustus 2017)*

--- Halaman ini sengaja dikosongkan ---