



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - EE 184801

**ANALISIS DAN IMPLEMENTASI KONTROL KECEPATAN
BLDC MOTOR MENGGUNAKAN *FUZZY LOGIC*
CONTROLLER PADA RODA DEPAN EV**

Sigit Armiyanto
NRP 0711154000071

Dosen Pembimbing
Dedet Candra Riawan, ST., M.Eng., Ph.D.
Heri Suryoatmojo, ST., MT., Ph.D.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2019



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - EE 184801

**ANALISIS DAN IMPLEMENTASI KONTROL KECEPATAN
BLDC MOTOR MENGGUNAKAN *FUZZY LOGIC*
CONTROLLER PADA RODA DEPAN EV**

Sigit Armiyanto
NRP 0711154000071

Dosen Pembimbing
Dedet Candra Riawan, ST., M.Eng., Ph.D.
Heri Suryoatmojo, ST., MT., Ph.D.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2019



ITS
Institut
Teknologi
Sepuluh Nopember

FINAL PROJECT - EE 184801

**ANALYSIS AND IMPLEMENTATION OF BLDC MOTOR
SPEED CONTROL USING FUZZY LOGIC CONTROLLER
FRONT IN-WHEEL IN EV**

Sigit Armiyanto
NRP 0711154000071

Supervisors
Dedet Candra Riawan, ST., M.Eng., Ph.D.
Heri Suryatmojo, ST., MT., Ph.D.

DEPARTMENT OF ELECTRICAL ENGINEERING
Faculty of Electrical Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2019

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa seluruh isi pada tugas akhir ini dengan judul "**ANALISIS DAN IMPLEMENTASI KONTROL KECEPATAN BLDC MOTOR MENGGUNAKAN FUZZY LOGIC CONTROLLER PADA RODA DEPAN EV**" adalah merupakan hasil karya intelektual mandiri, diselesaikan tanpa menggunakan sumber materi yang tidak diizinkan 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, Juli 2019



Sigit Armiyanto
0711154000071

**ANALISIS DAN IMPLEMENTASI KONTROL KECEPATAN
BLDC MOTOR MENGGUNAKAN FUZZY LOGIC
CONTROLLER PADA RODA DEPAN EV**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik
Pada
Bidang Studi Teknik Sistem Tenaga
Departemen Teknik Elektro
Institut Teknologi Sepuluh Nopember**

Menyetujui :

Dosen Pembimbing I



Dedet Candra Riawan, S.T., M.Eng., Ph.D.
NIP. 19731119 200003 1 001

Dosen Pembimbing II



Heri Suryoatmojo, ST., MT., Ph.D.
NIP. 19800603 200604 1 003

**SURABAYA
JULI, 2019**

ANALISIS DAN IMPLEMENTASI KONTROL KECEPATAN BLDC MOTOR MENGGUNAKAN *FUZZY LOGIC CONTROLLER* PADA RODA DEPAN EV

Nama : Sigit Armiyanto
Dosen Pembimbing I : Dedet Candra Riawan, S.T., M.Eng., Ph.D
Doaen Pembimbing II : Heri Suryoatmojo, ST., MT., Ph.D.

ABSTRAK

Berkembangnya kendaraan yang menggunakan bahan bakar fosil mengakibatkan lingkungan menjadi semakin tercemar dan berdampak pada kesehatan manusia. Banyak penelitian telah dikembangkan mengenai electric kendaraan, terutama pada komponen penggerak yang menyebabkan kerugian mekanis selama mode operasi. Seperti yang diketahui bahwa sistem penggerak pada kendaraan mempunyai model yang berbeda-beda, diantaranya adalah FF (Front engine-Front wheel drive), FR (Front engine-Rear wheel drive), MRW (Mid engine-Rear wheel drive), dan FWD/AWD (Four wheel drive / All wheel drive). Namun sistem penggerak masih menggunakan gearbox. Hal ini menyebabkan kerugian mekanis yang berdampak pada menurunnya efisiensi bahan bakar. Belakangan mulai dikembangkan system penggerak tanpa gearbox melainkan dengan *steering* motor Brushless DC. *Steering* yang digunakan yaitu menggunakan parameter kecepatan dan sudut. Sudut merupakan referensi mobil berjalan, saat dalam jalan lurus atau berbelok. Metode kontrol kecepatan motor Brushless DC menggunakan Pulse width modulation dan *feedback* kecepatan aktual roda dengan kontrol *fuzzy-logic*. Dengan *fuzzy-logic* maka kecepatan roda kanan dan kiri dapat diatur berjalan dengan kondisi yang diinginkan. Implementasi dari simulasi yang telah dilakukan pada tesis sebelumnya menampilkan data yang mendekati sama dengan hasil simulasi. Namun, masih terdapat eror dalam sensor pembacaan kecepatan aktual sehingga menyebabkan *feedback* eror yang dibandingkan dengan referensi menjadi besar. Hal ini mengakibatkan putaran roda mengalami nilai yang berubah-ubah. Dalam penelitian eror yang diakibatkan berkisar antara 0.2-1%.

Kata kunci : Motor Brushless DC, Fuzzy-Logic, PWM, Steering.

--- halaman ini sengaja dikosongkan ---

ANALYSIS AND IMPLEMENTATION OF BLDC MOTOR SPEED CONTROL USING FUZZY LOGIC CONTROLLER FRONT IN-WHEEL IN EV

Name : Sigit Armiyanto
Supervisor I : Dedet Candra Riawan, S.T., M.Eng., Ph.D
Supervisor I : Heri Suryoatmojo, ST., MT., Ph.D.

ABSTRACT

The development of vehicles using fossil fuels has resulted in the environment becoming increasingly polluted and affecting human health. Many studies have been developed regarding electric vehicles, especially in driving components that cause mechanical losses during operation mode. As is known that the driving system in vehicles has different models, including FF (Front engine-Front wheel drive), FR (Front engine-Rear wheel drive), MRW (Mid engine-Rear wheel drive), dan FWD/AWD (Four wheel drive / All wheel drive). But the driver system still uses the gearbox. This causes mechanical losses that have an impact on decreasing fuel efficiency. Later began to develop a drive system without a gearbox but with a Brushless DC steering motor. Steering used is using the speed and angle parameters. The angle is the reference for the car to run, when on a straight road or turn. The Brushless DC motor speed control method uses PWM and actual wheel speed feedback with fuzzy-logic controls. With fuzzy-logic, the right and left wheel speeds can be set to run with the desired condition.

This implementation from the simulations that have been carried out in the previous thesis displays data that is close to the simulation results. However, there is still an error in the actual speed from speed sensor which causes a large error feedback compared to the reference speed. This resulted in the wheel turning experiencing variable values. In the study errors caused ranged from 0.2 - 1%.

Keywords : Brushless DC Motor, Fuzzy-Logic, PWM, Steering.

--- halaman ini sengaja dikosongkan ---

KATA PENGANTAR

Puji Syukur atas keadirat Tuhan Yang Maha Esa dan segala rahmat yang telah dilimpahkan-Nya sehingga penulis mampu menyelesaikan tugas akhir dengan judul **“ANALISIS DAN IMPLEMENTASI KONTROL KECEPATAN BLDC MOTOR MENGGUNAKAN FUZZY LOGIC CONTROLLER PADA RODA DEPAN EV** dengan tepat waktu.

Tugas akhir ini disusun sebagai salah satu persyaratan untuk menyelesaikan pendidikan S1 pada Bidang Studi Teknik Sistem Tenaga, Departemen Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember. Atas selesainya pembuatan laporan tugas akhir ini, penulis mengucapkan terima kasih kepada:

1. Keluarga Penulis terutama ayah, ibu, yang telah merawat sejak dari kecil hingga sekarang serta selalu mendukung baik secara moral dan material. Kakak dan adik saya yang selalu menjadi penyemangat penulis dalam menyelesaikan tugas akhir.
2. Kementerian Riset, Teknologi, dan Pendidikan Tinggi yang telah memberikan beasiswa kepada penulis sehingga mampu melanjutkan pendidikan di perguruan tinggi.
3. Bapak Dr. Eng. Ardyono Priyadi, ST., M.Eng. selaku kepala Departemen Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember.
4. Bapak Dedet Candra Riawan, ST., M.Eng.,Ph.D dan Heri Suryoatmojo, ST., MT., Ph.D selaku dosen pembimbing yang telah memberikan arahan dan petunjuk dalam menyelesaikan laporan tugas akhir.
5. Seluruh dosen dan karyawan Departemen Teknik Elektro ITS yang telah memberikan banyak ilmu pada saat perkuliahan maupun diluar perkuliahan.
6. Mas chhith yang sudah meluangkan waktunya untuk membimbing dan memberikan arahan penulis.
7. Teman-teman e-55 khususnya lab elektronika B202, A206 dan lab Konversi Energi yang saling berjuang dan memberi semangat satu sama lain dalam pengerjaan tugas akhir.

Penulis menyadari bahwa terdapat banyak ketidaksempurnaan pada laporan tugas akhir ini, oleh karena itu diharapkan saran dari pembaca

agar nantinya laporan tugas akhir ini dapat menjadi referensi yang berguna bagi penelitian selanjutnya.

Surabaya, Juli 2019

Penulis

DAFTAR ISI

PERNYATAAN KEASLIAN TUGAS AKHIR	i
ABSTRAK	v
ABSTRACT	vii
KATA PENGANTAR	ix
DAFTAR GAMBAR	viii
DAFTAR TABEL.....	viv
BAB 1 PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Perumusan Masalah	2
1.3. Tujuan Tugas Akhir	2
1.4. Batasan Masalah	2
1.5. Metodologi	3
1.6. Sistematika Penulisan.....	4
1.7. Relevansi	5
BAB 2 ANALISIS DAN IMPLEMENTASI KONTROL KECEPATAN BLDC MOTOR MENGGUNAKAN <i>FUZZY LOGIC CONTROLLER</i> PADA RODA DEPAN <i>EV</i>	7
2.1. Sistem Pengemudian <i>Electric Vehicle</i>	7
2.1.1. Front Engine-Front Wheel Drive (FF).....	7
2.1.2. Front Engine-Rear Wheel Drive (FR)	8
2.1.3. Mid Engine-Rear Wheel Drive (MRW)	8
2.1.4. Four Wheel Drive / All Wheel Drive (FWD/AWD)	9
2.2. Independent Bldc Motor.....	10
2.2.1. Sistem Diferensial Elektronik (EDS).....	11
2.3. Motor Brushless DC.....	13
2.4. <i>Electronic Speed Control</i> (ESC).....	16
2.5. Metode Kontrol Tegangan ESC dengan PWM	20
2.5.1. Metode Bipolar	20
2.5.2. Metode Unipolar	21
2.6. Logika Fuzzy	22
2.6.1. Himpunan <i>Fuzzy</i>	23
2.6.2. Fungsi Keanggotaan (<i>Membership Function</i>).....	24

2.6.3. Operasi Himpunan <i>Fuzzy</i>	28
2.6.4. Sistem Inferensi <i>Fuzzy</i>	30
BAB 3 PERANCANGAN SISTEM KONTROL KECEPATAN BLDC MOTOR MENGGUNAKAN <i>FUZZY LOGIC CONTROLLER</i> PADA RODA DEPAN EV	33
3.1. Spesifikasi Motor <i>Brushless</i> DC	33
3.2. Sistem Kendali Kecepatan	34
3.2.1. Implementasi Ackermann-Jeantand Front-Wheel Drive	34
3.3. Sistem Drive pada <i>Brushless</i> DC	36
3.3.1. Program Kalibrasi ESC	36
3.3.2. Program Perhitungan Kecepatan Putaran	37
3.4. Program Kontrol Kecepatan BLDC dengan <i>Fuzzy- logic</i>	37
BAB 4 HASIL IMPLEMENTASI DAN PEMBAHASAN	43
4.1. Pengujian sensor kecepatan optocoupler	43
4.2. Pengujian tes PWM sebelum kalibrasi	44
4.3. Pengujian tes PWM setelah kalibrasi	44
4.4. Pengujian steering motor kontrol fuzzy.....	45
4.4.1. Sudut $\delta = 0$ (pada kondisi jalan lurus).....	45
4.4.2. Sudut $\delta = 15$ (pada kondisi belok kanan)	48
4.4.3. Sudut $\delta = -25$ (pada kondisi belok kiri).....	52
BAB V KESIMPULAN DAN SARAN	57
5.1. Kesimpulan	57
5.2. Saran	57
DAFTAR PUSTAKA.....	59
LAMPIRAN.....	61
Biodata Penulis.....	83

DAFTAR GAMBAR

Gambar 2.1. Skema Front Engine-Front Wheel Drive	7
Gambar 2.2 Skema Front Engine-Rear Wheel Drive	8
Gambar 2.3. Skema Mid Engine-Rear Wheel Drive	9
Gambar 2.4. Sekma 4WD/AWD	10
Gambar 2.5. Skema Pengaturan Kecepatan Independent Roda Depan .	10
Gambar 2.6. Model Pengaturan Ackermann-Jeantand	12
Gambar 2.7. Diagram Algoritma EDS.....	13
Gambar 2.8. Skema Rotasi Rotor 1	14
Gambar 2.9. Distribusi Fluks pada Fasa A.....	15
Gambar 2.10. Rangkaian Ekuivalen Motor BLDC	15
Gambar 2.11. Electronic Speed Control	17
Gambar 2.12. Contoh aliran arus pada fasa motor brushless	18
Gambar 2.13. Keluaran pembacaan sensor hall-effect	18
Gambar 2.14. Keluaran pembacaan EMF	19
Gambar 2.15. Metode Bipolar	20
Gambar 2.16. Metode Unipolar	20
Gambar 2.17. Sinyal output bipolar.....	21
Gambar 2.18. (A) upper switch PWM , (B) lower switch PWM	22
Gambar 2.19. Fungsi Keanggotaan Linear Naik	25
Gambar 2.20. Kurva Representasi Segitiga.....	26
Gambar 2.21. Kurva Representasi Trapesium.....	26
Gambar 2.22. Daerah ‘Bahu’ Pada Variabel Temperatur	28
Gambar 2.23. Operasi Himpunan Fuzzy.....	29
Gambar 3.1 Skema sistem kendali kecepatan	34
Gambar 3.2. Kendali kecepatan motor.....	34
Gambar 3.3. Blok Diagram Sistem Kontrol.....	38
Gambar 3.4. Membership function sinyal eror.....	39
Gambar 3.5. Defuzzifikasi singleton sistem kontrol.....	40
Gambar 3.6. Skema Flowchart Sistem Kontrol.....	41
Gambar 4.1 Grafik kecepatan sebelum kalibrasi.....	44
Gambar 4.2 Grafik kecepatan setelah kalibrasi.....	45
Gambar 4.3 Referensi kecepatan dengan sudut 0.....	46
Gambar 4.4 Kecepatan aktual dengan sudut 0	46
Gambar 4.5 Aktual dan referensi roda kiri dengan sudut 0	48
Gambar 4.6 Aktual dan referensi roda kanan dengan sudut 0.....	48
Gambar 4.7 Referensi kecepatan dengan sudut 15	49
Gambar 4.8 Kecepatan aktual dengan sudut 15	50

Gambar 4.9 Aktual dan referensi roda kiri dengan sudut 15	50
Gambar 4.10 Aktual dan referensi roda kanan dengan sudut 15	52
Gambar 4.11 Referensi kecepatan dengan sudut -25	53
Gambar 4.12 Kecepatan aktual dengan sudut -25.....	54
Gambar 4.13 Aktual dan referensi roda kiri dengan sudut -25.....	55
Gambar 4.14 Aktual dan referensi roda kanan dengan sudut -25	55

DAFTAR TABEL

Tabel 3.1. Spesifikasi Motor Brushless DC.....	33
Tabel 3.2. Rule-Base sistem kontrol.....	39
Tabel 4.1 Pengujian Sensor Kecepatan Optocoupler	43
Tabel 4.2 Error Pembacaan pada sudut 0.....	45
Tabel 4.3 Error Pembacaan pada sudut 15.....	47
Tabel 4.4 Error Pembacaan pada sudut -25	51

BAB 1

PENDAHULUAN

1.1. Latar Belakang

Berkembangnya kendaraan yang menggunakan bahan bakar fosil mengakibatkan lingkungan menjadi semakin tercemar dan berdampak pada kesehatan manusia [1]. Selain itu, desain pada kendaraan konvensional tergolong cukup lama, sehingga menimbulkan kecenderungan untuk mendapatkan desain yang lebih efisien dan juga menggunakan sumber yang ramah bagi lingkungan. Oleh karena itu, banyak penelitian yang telah dikembangkan mengenai *electric vehicle*, terutama pada komponen penggerak yang menyebabkan kerugian mekanis selama mode operasi.

Disisi lain, adanya Brushless DC motor yang merupakan salah satu motor penggerak tanpa menggunakan sikat dimana komutasinya secara elektrik berdasarkan posisi rotornya [2]. Motor Brushless DC sangat banyak diaplikasikan pada bidang industri karena tingkat efisiensi dan keandalannya yang tinggi, motor Brushless DC juga memiliki konstruksi sederhana, pemeliharaan yang mudah, dan memiliki keunggulan rasio/inersia yang tinggi [2]. Salah satu metode yang digunakan untuk mengatur kecepatan pada motor Brushless DC adalah dengan cara mengatur tegangan pada sisi *stator* menggunakan metode PWM. Keunggulan metode tersebut yaitu mempunyai struktur yang tidak rumit dan sudah umum diaplikasikan pada motor Brushless DC [3]. Variasi kecepatan didapatkan dengan mengatur *duty cycle* pada PWM. Diperlukan sistem kontrol untuk mengatur *duty cycle* pada PWM.

Seperti yang diketahui bahwa sistem penggerak pada kendaraan mempunyai model yang berbeda-beda, diantaranya adalah FF (Front engine-Front wheel drive), FR (Front engine-Rear wheel drive), MRW (Mid engine-Rear wheel drive), dan FWD/AWD (Four wheel drive / All wheel drive). Setiap model sistem penggerak mempunyai keunggulan dan juga kekurangan masing-masing. Namun, hal yang menjadi persamaan dari sistem penggerak adalah adanya gearbox. Hal ini menyebabkan kerugian mekanis yang berdampak pada menurunnya efisiensi bahan bakar. Prinsip yang digunakan pada pemodelan ini hampir sama dengan sistem FF, yaitu roda depan berfungsi sebagai penggerak. Hal ini dikarenakan penggerak roda depan memiliki keuntungan dapat menyudutkan roda lebih baik dan lebih mudah dikendalikan. Penggerak roda ini

menggunakan brushless dc motor yang dikendalikan secara independent antara roda kanan dan roda kiri. Berdasarkan kondisi tersebut, pada penelitian ini ditujukan untuk membuat sebuah model implementasi berdasarkan penelitian yang telah dilakukan sebelumnya mengenai sistem pengaturan kecepatan independent pada motor brushless yang diaplikasikan pada roda penggerak EV. [4] Sistem implementasi ini menggunakan metode fuzzy controler sebagai respon feedback kecepatan aktual motor yang kemudian dibandingkan dengan nilai set point yang telah ditentukan dan keluaran dan hasil perbandingan ini digunakan untuk mengatur besarnya PWM agar kedua roda dapat berputar sesuai kondisi yang diinginkan yaitu pada kondisi jalan yang lurus maupun berbelok.

1.2. Perumusan Masalah

Adapun perumusan masalah dalam penelitian ini adalah :

1. Mendesain prototype penggerak roda depan EV menggunakan motor Brushless DC
2. Mendesain program pengaturan tegangan motor Brushless DC menggunakan metode PWM dengan mikrokontroler.
3. Mendesain program kontrol kecepatan menggunakan fuzzy untuk mendapatkan feedback respon.
4. Membandingkan dan menganalisis hasil respon kecepatan antara sistem kontrol roda kanan dan roda kiri.

1.3. Tujuan Tugas Akhir

Dalam penelitian ini, terdapat tujuan dan manfaat adalah untuk mendapatkan prototype sistem implementasi dari penggerak roda depan EV menggunakan motor Brushless DC dengan metode fuzzy logic sebagai feedback respon untuk mengatur PWM motor yang kemudian akan dibandingkan respon roda kanan dan kiri sehingga dapat bergerak selaras.

1.4. Batasan Masalah

Pada penelitian ini, ditentukan batasan-batasan masalah sebagai berikut :

1. Motor brushless DC yang digunakan adalah A2212/13T
2. Sumber yang digunakan pada penelitian ini adalah baterai lipo 3 sel dengan kapasitas 1500 mAh.

3. Penggunaan *electric speed control* sebagai pengaturan kecepatan motor brushless DC.
4. Implementasi alat adalah sebuah prototype
5. Kontrol kecepatan menggunakan fuzzy

1.5. Metodologi

Adapun metodologi yang digunakan pada penelitian ini adalah sebagai berikut :

1. Studi Literatur

Dalam studi literatur, penulis mempelajari dasar-dasar tentang motor brushless DC, mencakup prinsip kerja, jenis-jenis motor, serta analisis-analisis terkait dengan brushless DC. Selain itu juga mempelajari fuzzy logic controller, pengaturan PWM, electric speed control dan sensor kecepatan. Literatur yang digunakan dalam studi literature ini diambil dari buku, jurnal ilmiah, prosiding, serta artikel

2. Identifikasi dan perancangan sistem

Pada tahap ini akan dilakukan identifikasi serta perancangan sistem yang akan dibuat. Selain itu akan dirancang sistem yang dapat digunakan untuk mengatur kecepatan dua Brushless DC agar selaras. Berikut merupakan tahap identifikasi dan perancangan sistem:

- a. Menentukan parameter-parameter pada sistem yang akan digunakan (Motor Brushless DC, tegangan suplai DC, ESC, Fuzzy Logic, sensor kecepatan).
- b. Penentuan nilai-nilai dari komponen tiap alat.
- c. Skema kontrol untuk mengatur PWM menggunakan mikrokontroler arduino

3. Simulasi Sistem

Tahap ini adalah mensimulasikan sistem secara keseluruhan. Dimana setelah menentukan nilai-nilai komponen dan merancang sistem, hasil dari perancangan tersebut dapat digunakan pada tahap simulasi. Simulasi akan dilakukan dengan menggunakan software MATLAB. Data yang diperoleh saat tahap identifikasi dan perancangan sistem akan dimasukkan ke tiap komponen sistem yang ada pada software, dan setelah itu simulasi dapat dilakukan

4. Analisis Data

Simulasi yang dilakukan akan menghasilkan data yang akan dianalisis seperti kecepatan yang dihasilkan brushless DC, respon logika fuzzy. Selain itu, dilakukan perbandingan dengan hasil perhitungan untuk mendapatkan hasil desain yang valid.

5. Implementasi Sistem

Simulasi yang telah dilakukan akan menentukan nilai-nilai yang sesuai dengan sistem control pada Brushless DC motor. Dan kemudian akan digunakan pada tahap perancangan prototype/hardware.

6. Penyusunan Laporan

Semua proses dan hasil penelitian ini akan ditulis dan disusun dalam laporan sebagai hasil penelitian.

1.6. Sistematika Penulisan

Sistematika penulisan dibagi menjadi lima bab yang terdiri dari :

- Bab 1. Pendahuluan
Bab ini membahas mengenai latar belakang masalah, permasalahan, tujuan, metodologi, sistematika penulisan, dan relevansi.
- Bab 2. Tinjauan Pustaka
Pada bab ini menjelaskan mengenai dasar teori dan teori pendukung terkait yang relevan terhadap tugas akhir ini. Dasar teori dan teori yang menunjang adalah seperti mengenai kontroler Fuzzy dan kontrol tegangan berbasis PWM, sistem EDS dan sensor kecepatan.
- Bab 3. Desain dan Permodelan Sistem
Pada bab ini akan dijelaskan bagaimana skema rancangan dari skema pengaturan kecepatan motor Brushless DC dengan metode PWM. Selain itu juga dijelaskan rancangan feedback system sebagai respon antara kecepatan actual dan referensi.
- Bab 4. Hasil Simulasi dan Pembahasan
Bab ini akan dibahas mengenai hasil perancangan sistem prototype dan analisis yang akan dibandingkan dengan perhitungan simulasi.
- Bab 5. Penutup
Pada bab terakhir ini akan dipaparkan kesimpulan dari penelitian dan saran untuk penelitian ini, agar kedepannya dapat

menjadi referensi yang berguna dalam pengembangan sistem kontrol kecepatan motor Brushless DC.

1.7. Relevansi

Hasil Penelitian tugas akhir ini memiliki hasil dengan harapan dapat memberikan manfaat berupa perkembangan dan inovasi, yaitu :

1. Sebagai referensi bagi peneliti lain yang ingin melakukan penelitian tentang sistem kontrol kecepatan motor Brushless DC berbasis Fuzzy.
2. Menjadi referensi kepada mahasiswa tentang penelitian Brushless DC khususnya tentang desain driver Brushless DC sebagai pengganti gearbox.

--- halaman ini sengaja dikosongkan ---

BAB 2

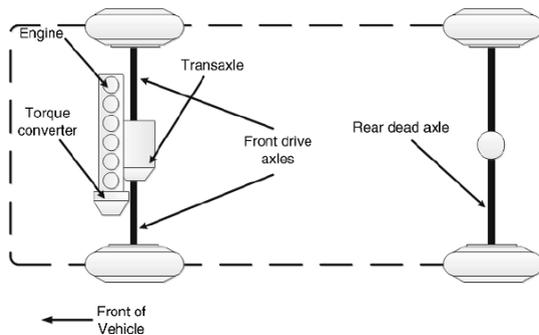
ANALISIS DAN IMPLEMENTASI KONTROL KECEPATAN BLDC MOTOR MENGGUNAKAN *FUZZY LOGIC* CONTROLLER PADA RODA DEPAN EV

2.1. Sistem Pengemudian *Electric Vehicle*

Sistem pengemudian mobil merupakan bagian yang paling penting. Sistem ini berfungsi untuk menggerakkan roda pada mobil agar dapat berjalan. Terdapat berbagai macam sistem penggerak roda pada kendaraan mobil, yaitu diantaranya adalah :

2.1.1. Front Engine-Front Wheel Drive (FF)

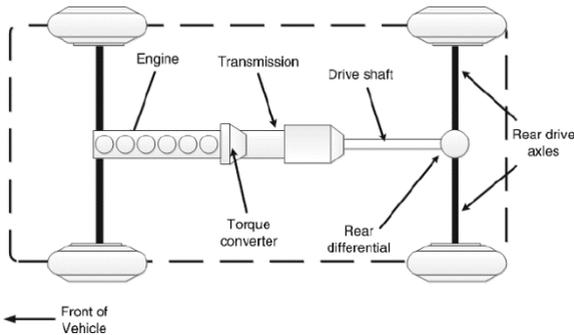
Front engine-front wheel drive atau biasa disingkat FF merupakan salah satu metode salah satu sistem penggerak roda, dimana roda bagian depan berfungsi untuk menggerakkan mobil. Komponen-komponen pada sistem penggerak jenis FF meliputi kopling(clutch), transmisi, gardan (*differential gearbox*) dan front axle. Kendaraan dengan menggunakan sistem FF memiliki kemampuan dalam menyudutkan roda lebih baik dengan metode yang lainnya, hal ini disebabkan karena mesin penggerak diletakkan diatas pengarah roda. Selain itu dapat meningkatkan efisiensi penghematan bahan bakar. Metode ini memiliki traksi yang rendah sehingga pada kondisi jalan berlumpur, basah, bersalju atau licin lebih baik daripada metode yang lainnya. Namun, apabila kecepatan pada kendaraan dinaikkan akan mengalami grip saat berbelok, sehingga berakibat understeer. Kurangnya pemindahan berat dapat membatasi akselerasi kendaraan roda depan.



Gambar 2. 1. Skema Front Engine-Front Wheel Drive

2.1.2. Front Engine-Rear Wheel Drive (FR)

Front Engine-Rear Wheel Drive (FR) merupakan penggerak mobil dengan cara memindahkan atau mentransfer tenaga dari mesin yang terdapat pada bagian depan ke roda belakang melalui gardan. Mesin RF memiliki desain yang sederhana dengan penempatan drive roda dibagian belakang sehingga dapat mengurangi kerumitan mekanis karena bagian transmisi daya diletakkan dibagian tengah. Komponen-komponen FR meliputi kopling (clutch), transmisi, drive shaft, gardan dan rear axle.



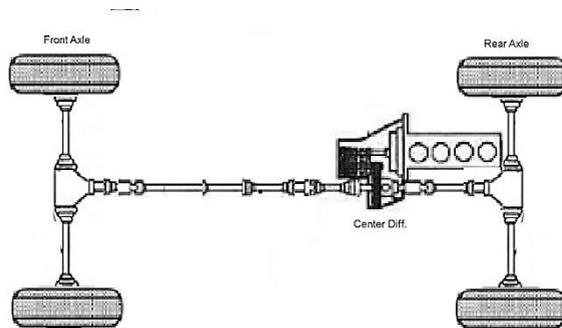
Gambar 2. 2. Skema Front Engine-Rear Wheel Drive

Mesin FR memiliki keuntungan akselerasi yang lebih tinggi dibandingkan FF, dikarenakan transfer berat beban ke bagian belakang kendaraan yang lebih baik. Keuntungan lainnya dari penempatan FR adalah kemudahan dalam mengakses ke kompartemen mesin, karena orientasi longitudinal drivetrain. Mobil FR cenderung lebih oversteer dibandingkan dengan FF. Namun, FR memiliki beberapa kekurangan yaitu sulit untuk dikendalikan dan penanganan yang lebih sulit pada cengkraman rendah seperti dalam kondisi jalan yang berlumpur, salju, dan basah.

2.1.3. Mid Engine-Rear Wheel Drive (MRW)

Mid Engine-Rear Wheel Drive (MR) memiliki konstruksi mesin yang terletak pada bagian tengah kendaraan dengan roda belakang sebagai penggerak. Bagian gardan pada MRW dapat diintegrasikan

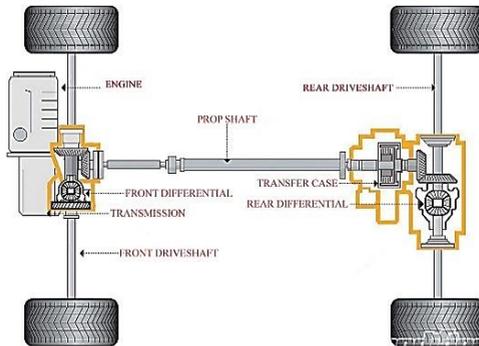
dengan sistem transmisi yang mengakibatkan drivetrain lebih efisien. Komponen-komponen pada sistem penggerak jenis MRW yaitu meliputi kopling (clutch), transmisi, gardan dan rear axle. Hal yang paling penting pada desain MRW adalah masalah stabilitas kendaraan. Ketika mesin berada di tengah, maka pemindahan berat cenderung dibebankan ke bagian belakang sehingga akan meningkatkan traksi pada roda depan dalam mengontrol pengereman kendaraan yang berguna untuk mengurangi kemungkinan tergelincirnya kendaraan.



Gambar 2. 3. Skema Mid Engine-Rear Wheel Drive

2.1.4. Four Wheel Drive / All Wheel Drive (FWD/AWD)

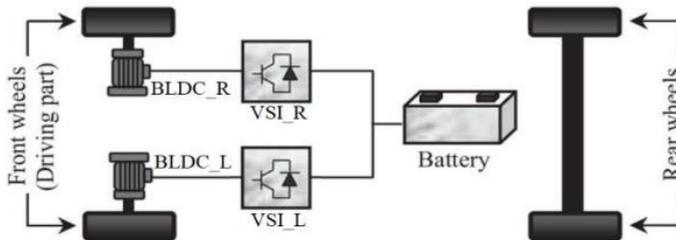
Four Wheel Drive/All Wheel Drive (4WD/AWD) merupakan sistem penggerak roda, dimana keempat rodanya dapat menerima torsi secara bersamaan. Komponen-komponen dari sistem pemindah tenaga jenis 4WD atau AWD meliputi kopling, transmisi, transfer case. Pada komponen transfer case ini dibagi menjadi 2 yaitu bagian yang menggerakkan roda depan yaitu meliputi front drive shaft, front differential dan front axle. Kemudian bagian yang satu lagi adalah yang menggerakkan roda belakang, yaitu meliputi rear drive shaft, rear differential dan rear axle.



Gambar 2. 4. Skema 4WD/AWD

2.2. Independent Bldc Motor

Independent BLCD motor merupakan salah satu metode yang dikembangkan sebagai penggerak roda mobil. Prinsip yang digunakan pada pemodelan ini hampir sama dengan sistem FF / FW (*Front Engine-Front wheel drive*), yaitu roda depan berfungsi sebagai penggerak. Hal ini dikarenakan penggerak roda depan memiliki keuntungan dapat menyudutkan roda lebih baik dan lebih mudah dikendalikan. Penggerak roda ini menggunakan bldc motor yang dikendalikan secara independent antara roda kanan dan roda kiri. Penggunaan bldc motor ditujukan untuk menghilangkan gearbox pada kendaraan yang mengakibatkan adanya rugi mekanis yang ditimbulkan sehingga efisiensi bahan bakar dapat lebih ditingkatkan.



Gambar 2. 5. Skema Pengaturan Kecepatan Independent Roda Depan

2.2.1. Sistem Diferensial Elektronik (EDS)

Pada pemodelan pengaturan independent roda depan menggunakan prinsip sistem differensial elektronik adalah untuk menunjukkan hubungan antara roda bagian dalam maupun bagian luar baik pada sistem roda dua atau roda empat. Dapat dilihat pada gambar skema sistem differensial menggunakan model Ackerman Jeantand.

Pada skema ini pengaturan kecepatan pada EV menggunakan pengendalian sudut δ , bagian dalam dan bagian luar pengaturan roda dapat direpresentasikan sebagai berikut :

$$\delta 1 = \arctan\left[\frac{L \tan \delta}{L - (\tan \delta \left(\frac{K}{2}\right))}\right] \dots\dots (2.1)$$

$$\delta 2 = \arctan\left[\frac{L \tan \delta}{L + (\tan \delta \left(\frac{K}{2}\right))}\right] \dots\dots (2.2)$$

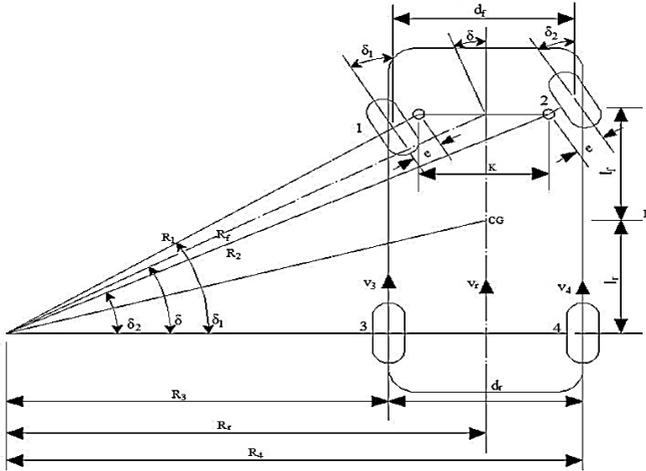
Dari gambar 6, dapat dihitung jarak untuk setiap roda adalah sebagai berikut :

$$R1 = \frac{L}{\sin(\delta 1)} \dots\dots\dots (2.3)$$

$$R2 = \frac{L}{\sin(\delta 2)} \dots\dots\dots (2.4)$$

Sedangkan untuk jarak pusat gravitasi kendaraan adalah :

$$R_{cg} = \sqrt{\left(\left(\frac{dr}{2}\right) + R3\right)^2 + \left(\frac{L}{2}\right)^2} \dots (2.5)$$



Gambar 2. 6. Model Pengaturan Ackermann-Jeantand

Dari rumus jarak antara setiap roda dan jarak pusat gravitasi kendaraan, maka dapat diperoleh kecepatan sudut untuk roda bagian dalam maupun luar selama kendaraan berputar sebagai berikut :

$$\omega_1 = \frac{V \cdot R_1}{R_{cg} \cdot r} \dots\dots\dots (2.6)$$

$$\omega_2 = \frac{V \cdot R_2}{R_{cg} \cdot r} \dots\dots\dots (2.7)$$

Dimana, L merupakan jarak antara roda depan dan roda belakang, K adalah jarak dari kiri ke kanan antar roda, r adalah jari-jari roda, V adalah kecepatan kendaraan.

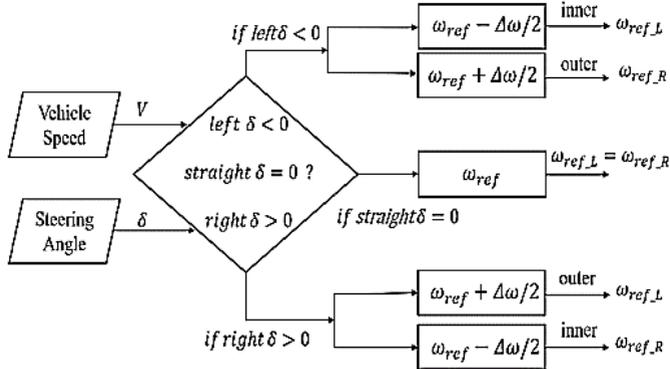
Terdapat beberapa kondisi yang dapat diasumsikan dalam pengaturan sudut, yaitu :

$$\begin{cases} \delta < 0, & \text{belok kiri} \\ \delta = 0, & \text{Lurus} \\ \delta > 0, & \text{belok kanan} \end{cases}$$

Perubahan kecepatan antara roda bagian dalam dan bagian luar dapat ditemukan sebagai berikut :

$$\Delta\omega = \frac{VL\left(\left(\frac{1}{\sin\delta_2}\right) - \left(\frac{1}{\sin\delta_1}\right)\right)}{r\sqrt{\left(\frac{L}{\tan\delta}\right)^2 + \left(\frac{L}{2}\right)^2}} \dots\dots(2.8)$$

Perubahan kecepatan ini digunakan untuk menambah atau mengurangi rata-rata kecepatan roda, ω_{ref} sebagai kecepatan sudut referensi pada sisi kanan dan sisi kiri.



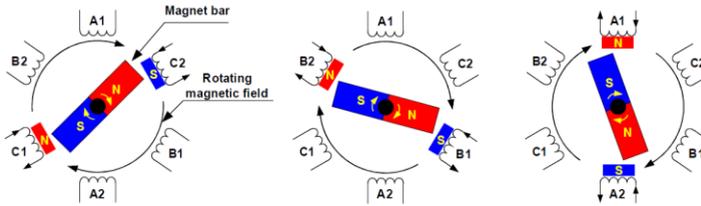
Gambar 2. 7. Diagram Algoritma EDS

2.3. Motor Brushless DC

Pada sistem pengaturan independent roda depan, motor yang digunakan adalah motor *brushless* DC. Motor *brushless* DC mempunyai persamaan dengan motor induksi dan motor *brushed* DC dalam hal konstruksi dan cara kerjanya. Motor brushless DC memiliki beberapa keuntungan dibandingkan dengan motor *brushed* DC atau motor induksi yaitu motor *brushless* memiliki keandalan yang tinggi, perawatan motor yang murah, noise yang rendah dan memiliki jangkauan kecepatan yang tinggi. Meskipun begitu, motor *brushless* DC juga memiliki kekurangan pada kontrol kecepatannya yang rumit dan harganya juga lebih mahal.

Motor *brushless* DC mempunyai bagian utama yaitu terdiri dari stator dan rotor. Motor *brushless* DC tiga fasa mempunyai tiga kumparan pada belitan stator. Dimana belitan stator terdapat 2 tipe kumparan yaitu *sinusoidal* dan *trapezoidal*. Kedua tipe tersebut dikategorikan berdasarkan bentuk dari sinyal BEMF (*Back Electromotive Force*). Bentuk dari sinyal BEMF ditentukan oleh perbedaan hubungan kumparan dan jarak dari celah udara. Motor brushless dengan bentuk sinyal BEMF *sinusoidal* mampu

menghasilkan torsi elektromagnetik lebih halus dibandingkan dengan bentuk sinyal BEMF *trapezoidal*.



Gambar 2. 8. Skema Rotasi Rotor 1

Pada rotor *brushless* DC terdapat shaft dan permanent magnet. Motor *brushless* DC memiliki prinsip kerja berdasarkan gaya tarik dan gaya lawan antara kutub magnet. Arus melewati salah satu dari kumparan stator, dan menghasilkan kutub magnet yang akan menarik kutub yang berlawanan dari magnet permanent yang terdekat. dengan secara bergantian mengalirkan arus pada kumpara stator, maka akan menyebabkan rotor akan berputar.

Pada *brushless* DC tegangan tiap fasa pada kumparan motor dapat dirumuskan dalam persamaan :

$$u_x = R_x i_x + e_{\psi_x} \quad (2.1)$$

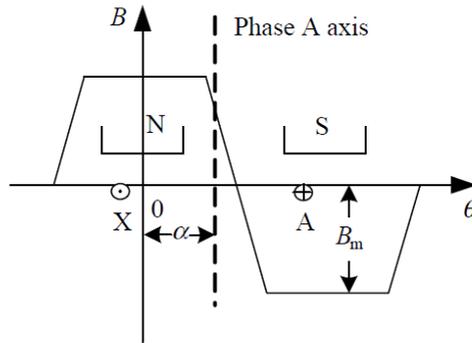
Dimana i , u , dan R merupakan parameter arus, tegangan, dan resistansi pada fasa x (fasa A, B, dan C). Sedangkan e_{ψ} adalah emf yang terinduksi pada fasa x . Besarnya emf yang terinduksi pada setiap fasa adalah sebanding dengan laju perubahan fluks.

$$e_{\psi_x} = \frac{d\psi_x}{dt} \quad (2.2)$$

Besarnya fluks pada fasa A adalah :

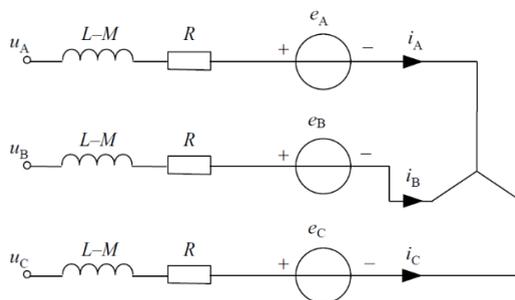
$$\psi_A = L_A i_A + M_{AB} i_B + M_{AC} i_C + \psi_{PM}(\theta), \quad (2.3)$$

Dimana ψ_{PM} merupakan *flux linkage* yang disebabkan oleh permanent magnet pada fasa A. Teta θ adalah sudut yang menyatakan posisi dari rotor, L_A adalah induktansi diri, M_{AB} dan M_{AC} adalah induktansi bersama fasa A dengan fasa B dan C. Besar dari $\psi_{PM}(\theta)$ bergantung pada distribusi medan magnet dari magnet permanen pada celah udara. Komponen radial dari medan magnet pada celah udara yang ditimbulkan oleh magnet permanen terdistribusi secara *trapezoidal* sepanjang permukaan dalam dari stator seperti yang ditunjukkan gambar 2.2.



Gambar 2. 9. Distribusi Fluks pada Fasa A

Pada motor Brushless DC 3 fasa memiliki rangkaian ekuivalen pada sisi stator adalah sebagai berikut :



Gambar 2. 10. Rangkaian Ekuivalen Motor BLDC

Pada rangkaian ekuivalen tersebut berlaku hukum arus,

$$i_A + i_B + i_C = 0 \quad (2.4)$$

Maka persamaan dapat disederhanakan menjadi,

$$u_A = Ri_A + (L - M) \frac{di_A}{dt} + e_A \quad (2.5)$$

Sehingga persamaan matriks tegangan fasa pada tiap kumpulan stator motor BLDC adalah sebagai berikut :

$$\begin{bmatrix} u_A \\ u_A \\ u_A \end{bmatrix} = \begin{bmatrix} R & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & R \end{bmatrix} \begin{bmatrix} i_A \\ i_A \\ i_A \end{bmatrix} + \begin{bmatrix} L - M & 0 & 0 \\ 0 & L - M & 0 \\ 0 & 0 & L - M \end{bmatrix} \frac{d}{dx} \begin{bmatrix} i_A \\ i_A \\ i_A \end{bmatrix} + \begin{bmatrix} e_A \\ e_A \\ e_A \end{bmatrix} \quad (2.6)$$

Sedangkan persamaan matriks untuk tegangan antar fasanya didapatkan dari pengurangan tegangan antar fasa dan didapatkan :

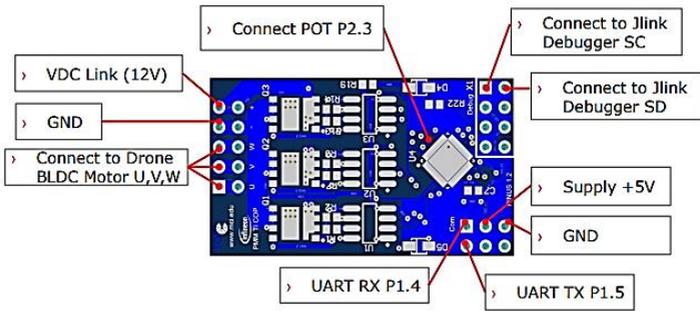
$$\begin{bmatrix} u_{AB} \\ u_{AC} \\ u_{CA} \end{bmatrix} = \begin{bmatrix} R & -R & 0 \\ 0 & R & -R \\ -R & 0 & R \end{bmatrix} \begin{bmatrix} i_A \\ i_A \\ i_A \end{bmatrix} + \begin{bmatrix} L - M & M - L & 0 \\ 0 & L - M & M - L \\ M - L & 0 & L - M \end{bmatrix} \frac{d}{dx} \begin{bmatrix} i_A \\ i_A \\ i_A \end{bmatrix} + \begin{bmatrix} e_A - e_B \\ e_B - e_C \\ e_C - e_A \end{bmatrix} \quad (2.7)$$

2.4. *Electronic Speed Control (ESC)*

Kontrol kecepatan elektronik mengikuti sinyal referensi kecepatan (berasal dari tuas throttle, joystick, atau input manual lainnya) dan memvariasikan tingkat switching dari *field effect transistor* (FET). Dengan menyesuaikan frekuensi switching dari transistor, kecepatan motor dapat diubah sesuai dengan referensi masukan. Pergantian frekuensi dari transistor inilah yang menyebabkan motor mengeluarkan suara bernada tinggi, terutama terlihat pada kecepatan yang lebih rendah.

Berbagai jenis kontrol kecepatan diperlukan untuk motor DC brushed dan motor DC brushless. Motor brushed memiliki kecepatan yang dapat dikontrol dengan memvariasikan tegangan pada angkernya. Sedangkan, motor brushless memerlukan prinsip operasi

yang berbeda. Kecepatan sepeda motor bervariasi dengan menyesuaikan waktu belitan motor saat ini.

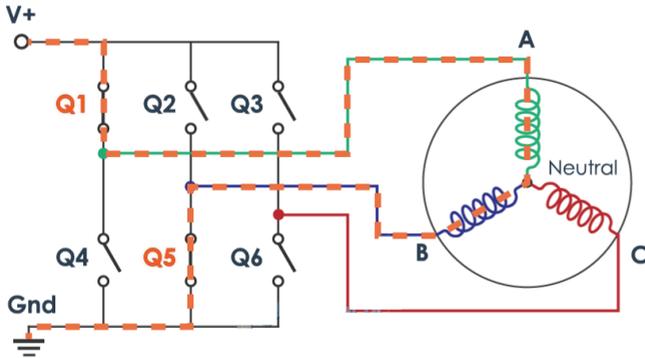


Gambar 2. 11. Electronic Speed Control

Sistem ESC brushless pada dasarnya menciptakan daya AC tiga fasa pada outputnya dengan cara pengaturan frekuensi, untuk menjalankan motor brushless. Pada ESC rotasi motor dideteksi dengan EMF atau dengan menggunakan hall sensor. Kontrol kecepatan pada ESC dapat diprogram melalui komputer dengan menggunakan mikrokontroler untuk menentukan batas tegangan rendah, pengaturan waktu, percepatan, pengereman, dan arah rotasi. Membalikkan arah motor juga dapat dilakukan dengan mengganti dua dari tiga output dari ESC ke motor.

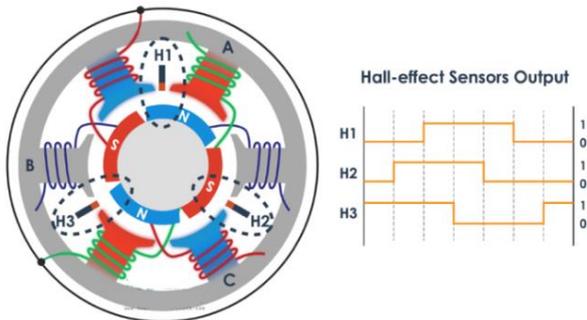
Rangkaian ESC terdiri dari switch yang telah diatur penyalannya sehingga menghasilkan sinyal AC pada sisi output. Sebagai contoh sistem penalaan dapat dilihat pada gambar 2.11. Ketika fase A pada motor Brushless DC dihubungkan ke sumber tegangan positif dan pada sisi lainnya fase B dihubungkan dengan ground, lalu untuk fasa C dalam kondisi floating, maka arus akan mengalir dari sumber tegangan positif melalui fase A menuju titik netral kemudian melewati fase B dan akhirnya menuju ground. Jadi dengan hanya menggunakan satu aliran arus, empat kutub yang berbeda dapat dihasilkan untuk menggerakkan rotor. Dengan konfigurasi seperti ini, akan terbentuk konfigurasi star pada fasa-fasa motor Brushless DC seperti gambar 2.11, dimana titik netral secara internal menghubungkan masing-masing ujung fasa dan ujung lainnya keluar dari motor. Hal ini menyebabkan mengapa motor brushless dc memiliki tiga kabel. Jadi untuk menghasilkan siklus penuh sehingga

dapat berputar, dua MOSFET yang benar harus diaktifkan pada setiap enam interval. Oleh karena itu dibutuhkan *electronic speed controller* (ESC).



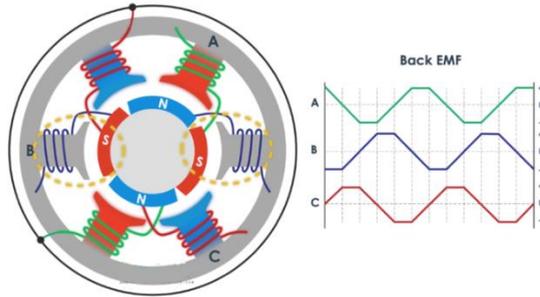
Gambar 2. 12. Contoh aliran arus pada fasa motor brushless

ESC berfungsi untuk mengendalikan kecepatan motor brushless dengan cara mengaktifkan MOSFET yang diperlukan untuk menghasilkan *rotating magnetic field* (RMF) sehingga motor dapat berputar. Semakin tinggi frekuensi masukan pada MOSFET, dengan kata lain semakin cepat ESC menempuh enam interval, maka semakin tinggi juga kecepatan motor Brushless DC yang akan dihasilkan. Untuk mengetahui fasa mana yang akan dilalui arus terlebih dahulu, dapat digunakan dengan cara mengetahui posisi rotor, terdapat dua metode umum yang digunakan untuk menentukan posisi rotor.



Gambar 2. 13. Keluaran pembacaan sensor hall-effect

Metode pertama adalah dengan menggunakan sensor *hall-effect* yang terletak pada stator dengan penempatan setiap 120 derajat atau 60 derajat antara satu sensor dengan sensor yang lainnya. Ketika rotor yang terdiri dari magnet permanen berputar, sensor *hall-effect* akan terinduksi medan magnet dan menghasilkan logika *high* untuk satu kutub dan *low* untuk kutub lainnya. Berdasarkan informasi ini ESC akan mengetahui saat untuk mengaktifkan interval komutasi selanjutnya.

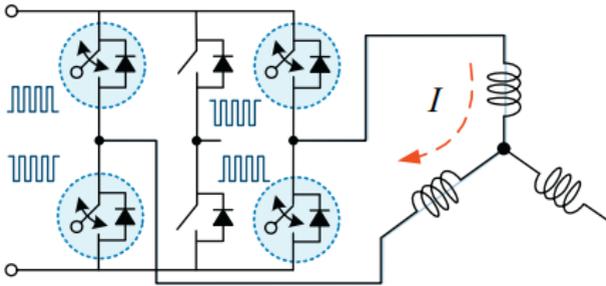


Gambar 2. 14. Keluaran pembacaan EMF

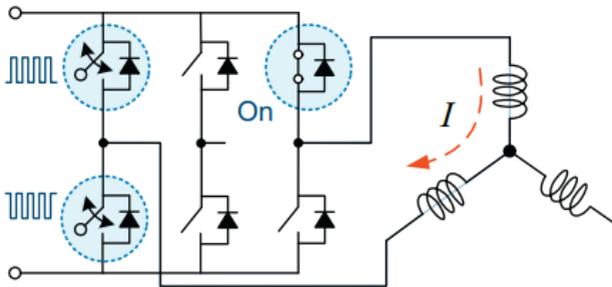
Metode kedua untuk menentukan posisi rotor adalah dengan menggunakan *electromotive force* balik yang dihasilkan, atau disebut sebagai *back EMF*. *Back EMF* dihasilkan oleh keballikan dari proses menghasilkan medan magnet, yaitu saat menggerakkan atau mengubah medan magnet yang melewati kumparan maka akan menginduksi arus pada kumparan tersebut. Jadi ketika medan magnet pada rotor bergerak melewati kumparan yang sedang tidak aktif, medan magnet ini akan menginduksi arus pada kumparan tersebut sehingga akan timbul beda tegangan. ESC akan menangkap beda tegangan ini saat terjadi dan berdasarkan hal tersebut ESC akan memperhitungkan kapan interval selanjutnya harus terjadi.

2.5. Metode Kontrol Tegangan ESC dengan PWM

Dalam drive motor BLDC, ada dua teknik PWM konvensional untuk menghasilkan tegangan seperti yang ditunjukkan pada Gambar 2.15 yaitu metode switching bipolar dan metode switching unipolar gambar 2.16. Prinsip ini mirip dengan teknik PWM dari rangkaian H-bridge untuk drive motor DC. Kesamaan ini disebabkan karena hanya terdapat dua fase yang diinduksikan sedangkan satu fasa yang lain memiliki kondisi floating dalam tiga fase inverter untuk drive motor Brushless DC.



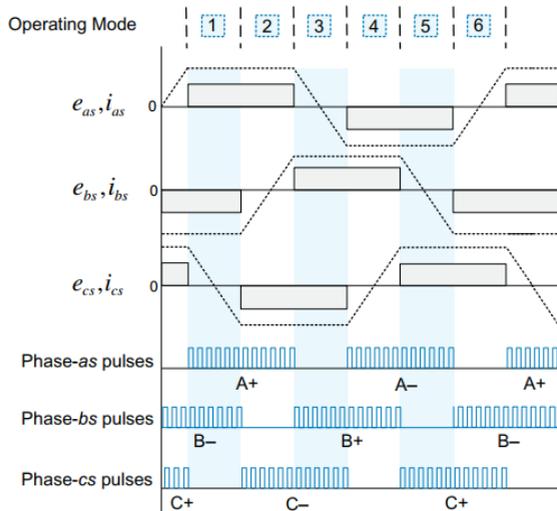
Gambar 2. 15. Metode Bipolar



Gambar 2. 16. Metode Unipolar

2.5.1. Metode Bipolar

Dalam metode switching bipolar, sinyal PWM diterapkan ke semua sakelar dari dua fase seperti yang ditunjukkan pada Gambar 2.15. Sebaliknya, dalam metode switching unipolar seperti yang ditunjukkan pada Gambar 2.16, sinyal PWM diterapkan pada sakelar hanya satu fase sementara satu sakelar fase lainnya disimpan pada keadaan aktif. Metode bipolar lebih sederhana dan dapat memberikan respons transien yang lebih baik karena $+V_{dc}$ atau $-V_{dc}$ diterapkan di seluruh fase berliku. Namun, riak torsi dan kehilangan switching lebih besar daripada metode switching unipolar. Sinyal gating PWM dengan metode switching bipolar ditunjukkan pada Gambar 2.17.

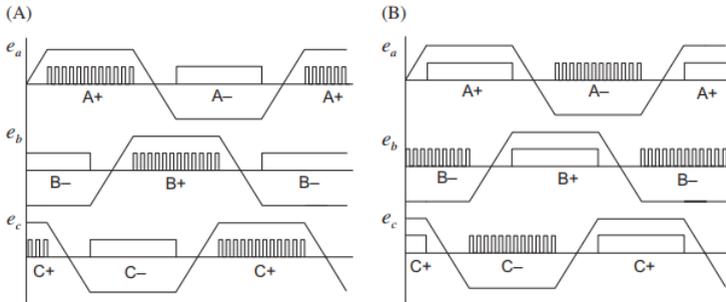


Gambar 2. 17. Sinyal output bipolar

2.5.2. Metode Unipolar

Dalam metode switching unipolar, kerugian switching dapat dikurangi karena sinyal PWM diterapkan ke switch hanya satu fase. Selain itu, karena tegangan yang diterapkan ke belitan fase adalah 0 dan +Vdc atau 0 dan -Vdc, riak arus adalah setengah dari metode perpindahan bipolar. Karena keunggulan ini, metode switching unipolar lebih banyak digunakan untuk drive motor BLDC. Namun, metode ini rumit dan memiliki respons yang lebih lambat daripada metode switching bipolar. Selain itu, arus sirkulasi dapat terjadi pada belitan fase tidak aktif. Dengan demikian metode ini kurang menguntungkan untuk drive servo presisi.

Metode switching unipolar juga dibagi menjadi beberapa skema sesuai dengan saklar yang mengeksekusi. Dalam skema PWM saklar atas pada Gambar 2.18A, sinyal PWM diterapkan pada saklar atas hanya sementara saklar bawah disimpan pada keadaan aktif. Di sisi lain, dalam skema PWM saklar bawah pada Gambar 2.18B, sinyal PWM diterapkan pada saklar bawah hanya sementara saklar atas disimpan pada keadaan aktif. Namun, dalam dua skema ini, pemanfaatan saklar dan rugi saklar bias karena hanya satu saklar khusus yang membuat saklar kontinu.



Gambar 2. 18. (A) upper switch PWM , (B) lower switch PWM

2.6. Logika Fuzzy

Kontrol kecepatan motor dapat menggunakan beberapa metode, salah satunya adalah dengan menggunakan *fuzzy-logic* kontroler. Logika fuzzy telah banyak digunakan pada sistem otomasi dan kontrol. Logika fuzzy dapat digunakan untuk menyelesaikan permasalahan nonlinier dan ketidakpastian tanpa menggunakan model matematika. Logika fuzzy merupakan suatu teori himpunan logika yang dikembangkan untuk mengatasi konsep nilai yang terdapat diantara nilai kebenaran (true) dan kesalahan (false). Logika fuzzy berbeda dengan logika digital biasa atau *Boolean*. Logika digital biasa hanya mengenal dua keadaan yang tegas (*crisp*), yaitu ‘ya’ atau ‘tidak’, ‘0’ atau ‘1’, dan ‘on’ atau ‘off’. Berbeda dengan logika digital biasa, logika fuzzy meniru cara berfikir manusia dengan menggunakan konsep kesamaan suatu nilai. Dengan menggunakan logika fuzzy, nilai tidak lagi hanya bernilai ‘0’ dan ‘1’ tetapi seluruh kemungkinan diantara 0 dan 1.

Kesederhanaan konsep membuat logika fuzzy mudah dimengerti. Fuzzy tidak terpaku dengan satu keputusan, fleksibel, sehingga dapat memberi nilai toleransi pada ketidakpastian. Ada beberapa alasan mengapa memilih menggunakan logika fuzzy yaitu :

1. Konsep logika fuzzy mudah dimengerti. Konsep matematis dari logika yang sangat sederhana.
2. Sifat logika fuzzy yang fleksibel
3. Logika fuzzy mampu menggambarkan fungsi-fungsi linier yang bersifat kompleks.

2.6.1. Himpunan *Fuzzy*

Pada himpunan crips, nilai keanggotaan hanya terdapat 2 kemungkinan yaitu '0' dan '1'. Sebagai contoh, terdapat sebuah elemen x dalam suatu himpunan A , yang dinyatakan dalam derajat keanggotaan $\mu_A(x)$, memiliki dua nilai keanggotaan, yaitu:

- a. Bernilai satu ($\mu_A(x) = 1$), yang berarti elemen x merupakan anggota penuh himpunan A ;
- b. Bernilai nol ($\mu_A(x) = 0$), yang berarti elemen x bukan merupakan anggota himpunan A .

Namun pada himpunan *fuzzy* yang menoleransi kesamaran, batas tegas yang memisahkan anggota dan bukan anggota pada himpunan tegas akan dieliminasi. Sehingga perpindahan antara anggota penuh dan bukan anggota akan terjadi berangsur-angsur karena keberadaan daerah *overlapping* pada himpunan *fuzzy*.

Terkadang kemiripan antara keanggotaan *fuzzy* dengan probabilitas menimbulkan kerancuan. Keduanya memiliki nilai pada interval $[0,1]$, namun interpretasi nilainya sangat berbeda antara kedua kasus tersebut. Keanggotaan *fuzzy* memberikan suatu ukuran terhadap pendapat atau keputusan, sedangkan probabilitas mengindikasikan proporsi terhadap keseringan suatu hasil bernilai benar dalam jangka panjang. Misalnya, jika nilai keanggotaan suatu himpunan *fuzzy* MUDA adalah 0,9; maka tidak perlu dipermasalahkan berapa seringnya nilai itu diulang secara individu untuk mengharapkan suatu hasil yang hampir pasti muda. Di sisi lain, nilai probabilitas 0,9 muda berarti 10% dari himpunan tersebut diharapkan tidak muda.

Himpunan *fuzzy* memiliki 2 atribut, yaitu :

1. Linguistik, yaitu penamaan suatu grup yang mewakili suatu keadaan atau kondisi tertentu dengan menggunakan bahasa alami, seperti: MUDA, PAROBAYA, TUA
2. Numeris, yaitu suatu nilai (angka) menunjukkan ukuran dari suatu variabel seperti : 10, 25, 50 dan sebagainya.

Berikut beberapa hal yang perlu diketahui mengenai sistem *fuzzy*, yaitu:

1. Variabel *fuzzy*

Variabel *fuzzy* merupakan variabel yang dibahas dalam sistem *fuzzy*. Contoh: temperatur, kecepatan, posisi, dll.

2. Himpunan *fuzzy*

Himpunan *fuzzy* merupakan suatu kelompok yang mewakili suatu keadaan tertentu dalam variabel *fuzzy*. Umumnya ciri penamaan dalam himpunan *fuzzy* adalah linguistik. Linguistik merupakan penamaan yang nilainya berupa kata-kata, yang digunakan untuk memperkirakan nilai atau keadaan yang cukup sulit untuk dinyatakan secara eksak. Contoh: lambat, cepat, sangat cepat.

3. Semesta pembicaraan

Semesta pembicaraan adalah keseluruhan nilai yang diperbolehkan untuk dioperasikan dalam suatu variabel *fuzzy*. Semesta pembicaraan merupakan himpunan bilangan real yang nilainya selalu bertambah secara monoton dari kiri ke kanan. Nilai semesta pembicaraan dapat berupa bilangan positif maupun bilangan negatif.

4. Domain

Domain himpunan *fuzzy* adalah keseluruhan nilai yang diijinkan dalam semesta pembicaraan dan boleh dioperasikan dalam suatu himpunan *fuzzy*. Sama seperti semesta pembicaraan, domain merupakan himpunan bilangan real yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan. Domain dapat berupa bilangan positif maupun bilangan negatif.

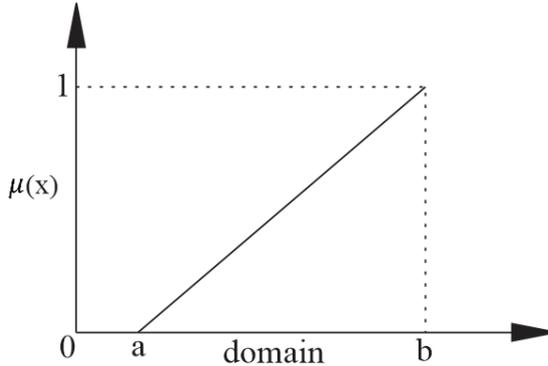
2.6.2. Fungsi Keanggotaan (*Membership Function*)

Fungsi keanggotaan (*membership function*) adalah suatu kurva yang menunjukkan pemetaan titik-titik *input* data ke dalam nilai keanggotaannya yang memiliki interval nilai antara 0 sampai 1. Salah satu cara yang digunakan untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi. Berikut beberapa fungsi keanggotaan yang umum digunakan, yaitu :

2.6.2.1. Representasi linier

Pada representasi linear, pemetaan input ke derajat keanggotaannya digambarkan sebagai garis lurus. Bentuk ini paling sederhana dan menjadi pilihan yang baik untuk mendekati suatu konsep yang kurang jelas.

Ada 2 keadaan himpunan fuzzy yang linear. Pertama, kenaikan himpunan dimulai pada nilai domain yang memiliki derajat keanggotaan nol [0] bergerak ke kanan menuju ke nilai domain yang memiliki derajat keanggotaan lebih tinggi.



Gambar 2. 19. Fungsi Keanggotaan Linear Naik

Fungsi keanggotaan :

$$\mu[x] = \begin{cases} 0; & x \leq 0 \\ \frac{(x-a)}{(x-b)}, & a \leq x \leq b \\ 1; & x \geq b \end{cases} \quad (2.14)$$

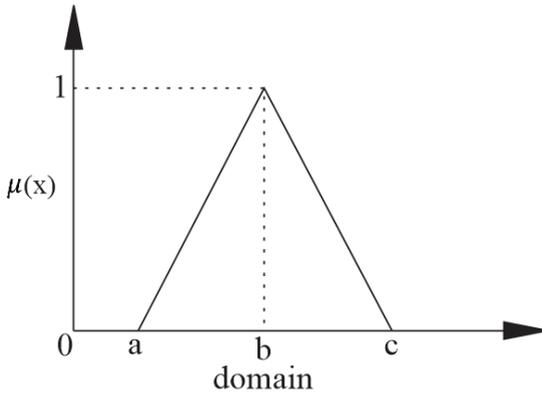
Kedua merupakan kebalikan yang pertama. Garis lurus di mulai dari nilai domain dengan derajat keanggotaan tertinggi pada sisi kiri, kemudian bergerak menurun ke nilai domain yang memiliki derajat keanggotaan lebih rendah.

$$\mu[x] = \begin{cases} \frac{(b-x)}{(b-a)}, & a \leq x \leq b \\ 0, & x \geq b \end{cases} \quad (2.15)$$

2.6.2.2. Representasi kurva segitiga

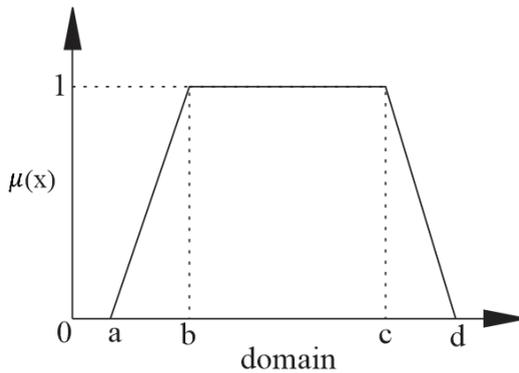
Fungsi segitiga merupakan gabungan antara dua fungsi linea seperti terlihat pada gambar. keanggotaan fungsi segitiga, yaitu:

$$\mu [x] = \begin{cases} 0 & ; x \leq a \\ \frac{(x-a)}{(b-a)} & ; a \leq x \leq b \\ \frac{(b-x)}{(c-b)} & ; b \leq x \leq c \\ 0 & , x \geq c \end{cases} \quad (2.16)$$



Gambar 2. 20. Kurva Representasi Segitiga

2.6.2.3. Representasi kurva trapesium



Gambar 2. 21. Kurva Representasi Trapesium

Kurva trapesium pada dasarnya seperti bentuk segitiga, dimana terdapat titik yang memiliki nilai keanggotaan 1.

Fungsi keanggotaan :

$$\mu [x] = \begin{cases} 0; & x \leq a \\ \frac{(x-a)}{(b-a)} & ; a \leq x \leq b \\ 1; & b \leq x \leq c \\ \frac{(c-x)}{(d-c)} & ; c \leq x \leq d \\ 0, & x \geq d \end{cases} \quad (2.17)$$

2.6.2.4. Representasi kurva bentuk bahu

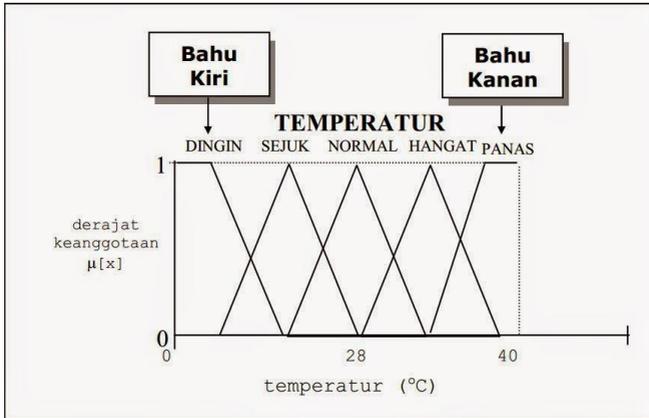
Daerah yang terletak di tengah-tengah suatu variabel yang direpresentasikan dalam bentuk segitiga, pada sisi kanan dan kirinya akan naik dan turun. Tetapi terkadang salah satu sisi dari variabel tersebut tidak mengalami perubahan. Sebagai contoh, apabila telah mencapai kondisi PANAS, kenaikan temperatur akan tetap berada pada kondisi PANAS. Himpunan fuzzy ‘bahu’, bukan segitiga, digunakan untuk mengakhiri variabel suatu daerah fuzzy. Bahu kiri bergerak dari 1 ke 0, sementara bahu kanan bergerak dari 0 ke 1. Gambar menunjukkan variabel TEMPERATUR dengan daerah bahunya.

Fungsi keanggotaan :

$$\mu_{dingin}[x] = \begin{cases} 1; & a \leq x \leq b \\ \frac{(c-x)}{(c-b)}, & b \leq x \leq c \\ 0; & x \geq c \end{cases} \quad (2.18)$$

$$\mu_{sejuk}[x] = \begin{cases} 0; & x \leq b \\ \frac{(x-b)}{(c-b)}, & b \leq x \leq c \\ \frac{(d-x)}{(d-c)}, & c \leq x \leq d \\ 0; & x \geq d \end{cases} \quad (2.19)$$

$$\mu_{normal}[x] = \begin{cases} 0; & x \leq c \\ \frac{(x-c)}{(d-c)}, & c \leq x \leq d \\ \frac{(e-x)}{(e-d)}, & d \leq x \leq e \\ 0; & x \geq e \end{cases} \quad (2.20)$$



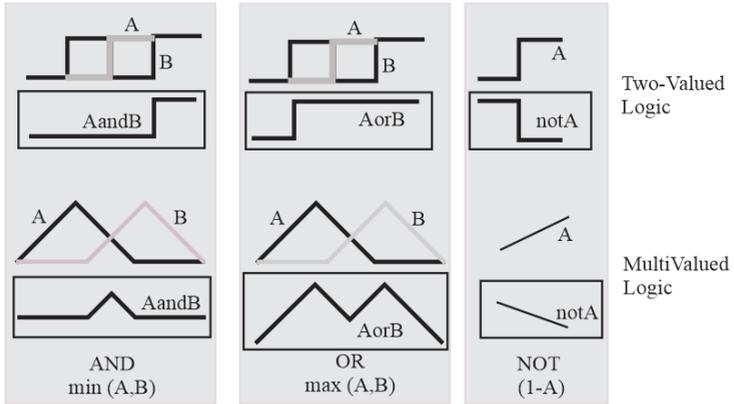
Gambar 2. 22. Daerah ‘Bahu’ Pada Variabel Temperatur

$$\mu_{hangat}[x] = \begin{cases} 0; & x \leq d \\ \frac{(x-d)}{(e-d)}, & d \leq x \leq e \\ \frac{(f-x)}{(f-e)}, & e \leq x \leq f \\ 0; & x \geq f \end{cases} \quad (2.21)$$

$$\mu_{panas}[x] = \begin{cases} 0; & x \leq e \\ \frac{(x-e)}{(f-e)}, & e \leq x \leq f \\ 1; & x \geq b \end{cases} \quad (2.22)$$

2.6.3. Operasi Himpunan Fuzzy

Berikut akan dijelaskan beberapa operator dasar yang umum digunakan untuk menggabungkan dan memodifikasi dua atau lebih himpunan fuzzy pada fungsi keanggotaan, yaitu operator interseksi, gabungan, dan komplemen. Masing-masing hasil operasi ditunjukkan oleh Gambar, kemudian akan dijelaskan sebagai berikut:



Gambar 2. 23. Operasi Himpunan Fuzzy

a. Operator interseksi (*AND*)

Interseksi himpunan *fuzzy* Adan himpunan *fuzzy* Byang didefinisikan dalam semesta pembicaraan U , dinotasikan sebagai $A \cap B$, memiliki dua nilai keanggotaan yaitu:

1. *Minimum*

Nilai keanggotaan *minimum* yaitu:

$$\mu_{A \cap B} = \min \{ \mu_A(u_i), \mu_B(u_i) : u_i \in U_i \} \quad (2.23)$$

2. *Algebraic product*

Nilai keanggotaan *product* yaitu:

$$\mu_{A \cap B} = \{ \mu_A(u_i) \mu_B(u_i) : u_i \in U_i \} \quad (2.24)$$

b. Operator gabungan (*OR*)

Gabungan himpunan *fuzzy* Adan himpunan *fuzzy* Byang didefinisikan dalam semesta pembicaraan U , dinotasikan sebagai $A \cup B$, memiliki dua nilai keanggotaan yaitu:

1. *Maximum*

Nilai keanggotaan *maximum* yaitu:

$$\mu_{A \cup B} = \max \{ \mu_A(u_i), \mu_B(u_i) : u_i \in U_i \} \quad (2.24)$$

2. Algebraic Sum

Nilai keanggotaan *algebraic sum* yaitu:

$$\mu_{A \cup B} = \begin{cases} \mu_A(u_i) + \mu_B(u_i) \\ - \mu_A(u_i) \mu_B(u_i) : u_i \\ \in U_i \end{cases} \quad (2.25)$$

c. Operator komplemen (*NOT*)

Komplemen himpunan *fuzzy A* dengan fungsi keanggotaan $\mu_A(u_i)$, memiliki nilai keanggotaan yaitu:

$$\mu_{A'}(u_i) = 1 - \mu_A(u_i) \quad (2.26)$$

2.6.4. Sistem Inferensi *Fuzzy*

Inferensi *fuzzy* adalah cara penarikan kesimpulan pada sistem *fuzzy*. Dalam penalaran logika *fuzzy*, tipe yang umum digunakan untuk inferensi yaitu *generalised modus ponens* (GMP) atau disebut juga *direct reasoning*. Karena pada umumnya pemikiran manusia dilakukan dengan cara *modus ponens*. Pemetaan *input* ke *output* pada sistem *fuzzy* berada dalam sekumpulan aturan *kondisi* \rightarrow *aksi* atau bentuk *IF-THEN* yang menggambarkan *modus ponens*, yaitu:

IF *premis* **THEN** *konsekuen*.

Premis merupakan kondisi dan konsekuen merupakan aksi. Umumnya *input* sistem *fuzzy* dihubungkan dengan premis sedangkan *output* sistem *fuzzy* dihubungkan dengan konsekuen.

Sistem *fuzzy* dibangun oleh empat bagian utama seperti yang ditunjukkan pada Gambar 2.12, yaitu:

1. Dasar-aturan

Dasar-aturan merupakan sekumpulan aturan *IF-THEN* yang berisi kuantifikasi uraian linguistik pakar tentang cara mencapai kontrol yang baik.

2. Mekanisme inferensi

Disebut juga inferensi *fuzzy*, yang menyamai cara pengambilan keputusan pakar dalam menerjemahkan dan mengaplikasikan pengetahuan tentang cara terbaik untuk mengontrol *plant*.

3. Fuzifikasi

Mengonversi *input* kontroler kedalam bentuk informasi yang dapat dipahami oleh mekanisme inferensi untuk mengaktifkan dan mengaplikasikan aturan-aturan *IF-THEN*.

4. Defuzifikasi

Mengonversi kesimpulan yang dihasilkan mekanisme inferensi menjadi *input* aktual untuk proses.

Terdapat beberapa langkah dalam mekanisme inferensi *fuzzy*, yaitu:

1. Fuzifikasi *input*

Mengonversi *input* kontroler yang merupakan *input* tegas menjadi *input fuzzy* lalu memetakannya kedalam fungsi keanggotaan himpunan *fuzzy* dengan derajat keanggotaannya masing-masing.

2. Aplikasi operator *fuzzy*

Mengombinasi dan memodifikasi ketika terdapat dua atau lebih himpunan *fuzzy* pada bagian premis.

3. Aplikasi metode implikasi

Menghasilkan sebuah *output* konsekuen untuk masing-masing aturan.

4. Agregasi semua *output*

Mengumpulkan semua *output* konsekuen aturan yang dihasilkan pada implikasi.

5. Defuzifikasi

Mengonversi *output fuzzy* menjadi *output* tegas sebagai *input* proses selanjutnya.

--- halaman ini sengaja dikosongkan ---

BAB 3

PERANCANGAN SISTEM KONTROL KECEPATAN BLDC MOTOR MENGGUNAKAN *FUZZY LOGIC CONTROLLER* PADA RODA DEPAN EV

3.1. Spesifikasi Motor *Brushless* DC

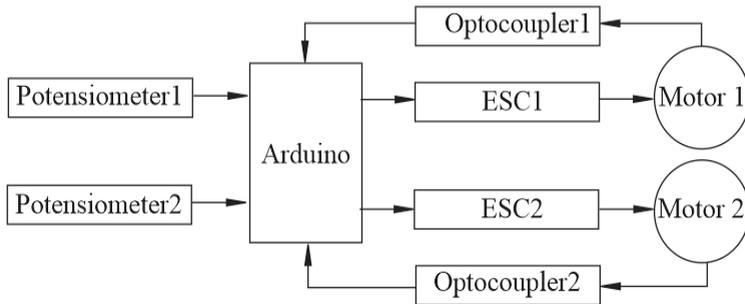
Motor brushless DC merupakan jenis motor yang paling sering diaplikasikan pada akhir-akhir ini. Hal ini dikarenakan motor brushless DC memiliki kelebihan dibandingkan dengan motor DC brushed. Rugi-rugi gesekan pada motor brushless lebih kecil karena tanpa menggunakan sikat. Akan tetapi motor brushless mempunyai kelemahan yaitu pada sistem kontrol kecepatannya yang rumit karena menggunakan komutasi elektrik. Motor brushless pada umumnya disuplai menggunakan baterai yang kemudian akan diubah menjadi gelombang AC oleh electric speed control sebelum masuk ke dalam motor.

Dalam penelitian ini motor brushless yang digunakan adalah A2212/13T.

Tabel 3.1. Spesifikasi Motor Brushless DC

Source	2-3 LiPo Battery
Kv	1000 RPM/V
Max Efficiency	80%
Max Efficiency current	4 – 10 A (>75%)
No Load Current	0.5A @10V
Resistance	0.090 ohm
Max Current	13A for 60S
Max Watts	150W
Weight	52.7 g/ 1.86 oz
Size	28 mm dia x 28 mm bell length
Shaft Diameter	3.2 mm
Poles	14
Model weight	300 – 800g / 10.5 – 28.2 oz

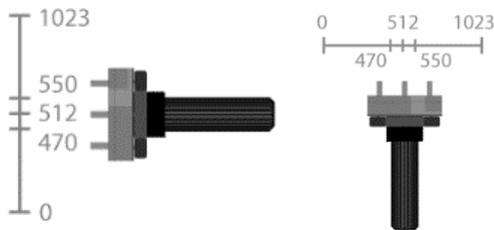
3.2. Sistem Kendali Kecepatan



Gambar 3. 1. Skema sistem kendali kecepatan

3.2.1. Implementasi Ackermann-Jeantand Front-Wheel Drive

Pada gambar 2.6 telah dijelaskan mengenai skema pemodelan steering dengan metode ackerman-jeantand. Dijelaskan bahwa steering pada motor dipengaruhi oleh dua parameter yaitu referensi kecepatan dan sudut. Dimana sudut akan mempengaruhi kecepatan putar roda kanan dan roda kiri dalam berbagai kondisi, saat berbelok maupun jalan lurus.



Gambar 3. 2. Kendali Kecepatan Motor menggunakan potensiometer

Pada implementasi sistem kendali kecepatan motor, menggunakan dua buah potensiometer yang telah diset sebagai fungsi dari kecepatan referensi dan juga steering sudut.

Pertama adalah digunakan untuk steering kecepatan. potensiometer terhubung ke input analog Arduino. Potensiometer memiliki nilai analog dari 0 hingga 1023. Pada pengaturan referensi kecepatan ini, potensiometer di set dari nilai 0 – 180 sebagai data input digital yang sudah dikonversi dari sebelumnya yang nilainya analog dari 0 hingga 1023. Input digital ini kemudian menjadi nilai PWM yang meningkat dari 0 menjadi 180 yang sebenarnya adalah kecepatan motor.

Hampir sama dengan potensiometer yang pertama, potensiometer kedua ditentukan sebagai fungsi steering sudut. Dimana nilai potensiometer yang telah diperlihatkan pada gambar 3.2, nilai analognya kemudian dikonversi menjadi fungsi sudut dari -45 derajat hingga 45 derajat. Nilai 0 derajat pada potensiometer diberikan interval dari 470 – 550 agar pada saat motor bergerak dengan sudut yang sama, steering sudut mampu menunjukkan angka 0 derajat. Karena jika tidak diberikan interval maka apabila potensiometer akan sulit untuk diatur menuju nilai 0 derajat dan cenderung langsung berubah tanda. Untuk sudut negative diatur dari interval 0 – 477 dan kemudian untuk nilai sudut positif diatur pada interval 550 – 1023.

Berdasarkan algoritma pada gambar 2.7, dijelaskan bahwa saat sudut steering 0 maka roda kanan dan roda kiri memiliki kecepatan yang sama atau dalam kondisi realnya adalah jalan lurus. Pada kondisi belok, sudut akan mempengaruhi kecepatan roda kanan dan kiri. Misalnya, saat sudut positif maka kondisi roda bagian kiri akan lebih cepat dibandingkan dengan roda sebelah kanan. Atau dalam kondisi realnya akan belok ke arah kanan. Begitu pula sebaliknya, saat kondisi sudut negatif maka roda kanan akan berputar lebih cepat dibandingkan roda sebelah kiri atau dalam kondisi realnya belok ke kiri. Dalam penerapan implementasi program arduino, output yang dihasilkan adalah sebuah nilai PWM yang dikirim ke ESC untuk menggerakkan motor. Dalam program ini, ketika sudut 0 maka nilai PWM pada roda kanan sama dengan roda kiri. Apabila sudut positif maka nilai PWM roda kiri akan bertambah sedangkan nilai PWM roda kanan akan berkurang. Begitu pun saat sudut negatif. Maka nilai PWM roda kanan akan bertambah sedangkan PWM roda kiri akan

berkurang. Nilai PWM pada salah satu roda akan semakin besar apabila sudut steeringnya juga semakin besar.

Pada implementasi ini, nilai PWM yang akan dikirimkan dibatasi nilai minimumnya sebesar 10. Hal ini diakibatkan motor yang digunakan tidak dapat berjalan saat nilai PWM di bawah 10.

3.3. Sistem Drive pada Brushless DC

Pada bagian ini akan dijelaskan mengenai sistem kendali kecepatan Brushless DC dengan PWM. Implementasi pemrograman pada arduino menggunakan library motor servo. Hal ini dipilih karena dengan menggunakan library motor servo maka akan lebih mudah dalam membangkitkan sinyal output dengan frekuensi 50Hz. Program ini bertujuan untuk menjaga duty cycle tetap pada angka 50% sedangkan PWM ini akan mengubah nilai amplitudanya. Semakin tinggi nilai PWM yang dibangkitkan, maka akan semakin tinggi pula amplitudanya yang mengakibatkan kecepatan motor akan semakin besar.

Kendali pw mini menggunakan sinyal analog yang dikirimkan oleh potensiometer yang kemudian diubah dalam nilai digital oleh arduino sebelum dikirimkan menuju ESC. Nilai dari potensiometer adalah 0-1023 dalam bentuk analog akan diubah menjadi 0-180 dalam bentuk digital output.

ESC tidak akan merespon PWM apabila inputnya langsung dalam nilai tertentu. Oleh karena itu, perlunya nilai output PWM dari arduino yang masuk ke dalam ESC diinisiasikan dimulai dari nilai 0. Dan nilai pw mini akan bertambah apabila nilai dari potensiometer juga bertambah.

3.3.1. Program Kalibrasi ESC

Pada ESC, nilai awal (inisialisasi) PWM yang dimiliki belum tentu sama sejak awal dibeli. Hal ini akan mengakibatkan perbedaan yang cukup signifikan antara putaran motor satu dengan yang lain ketika arduino mengirimkan sinyal kepada ESC meskipun dengan nilai yang sama. Oleh karena itu, perlu adanya proses kalibrasi agar nilai yang dimiliki antara ESC menjadi sama atau berdekatan. Berikut adalah program untuk mengkalibrasi ESC.

Sistem kerja program diatas adalah dengan memberikan beberapa nilai PWM yang berbeda-beda. Dengan perintah “flag” yang berfungsi untuk mengunci suatu nilai PWM pada kondisi tertentu. Oleh karena itu, saat ESC A diberikan nilai PWM tertentu maka command “flag” akan memberhentikan nilai ESC A dalam rentan waktu tertentu. Dan pada saat yang sama, ESC B juga akan mempunyai nilai PWM yang sama karena nilainya sudah diblok dalam range tertentu. Kemudian nilai ESC A dan B dikembalikan menjadi 0 dalam keadaan semula dan kemudian nilai ini diblok sehingga nilainya menjadi sama.

3.3.2. Program Perhitungan Kecepatan Putaran

Pada saat motor diberikan sinyal dari arduino, maka yang masuk kedalam motor adalah dalam bentuk PWM bukan nilai rpm. Oleh karena itu, diperlukan sensor untuk membaca nilai putaran yang disebabkan oleh PWM dari output arduino. Nilai ini akan digunakan dalam pembuatan sistem kontrol menggunakan fuzzy-logic.

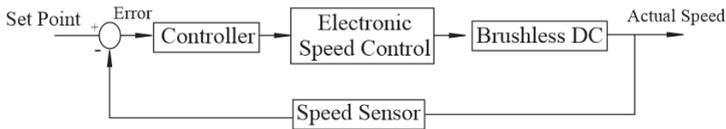
3.4. Program Kontrol Kecepatan BLDC dengan Fuzzy-logic

Setelah didapatkan nilai rpm pada motor, maka nilai kecepatan ini akan dibandingkan dengan sinyal referensi yang dikirimkan arduino ke dalam ESC. Logika fuzzy ini terjadi didalam arduino yang kemudian diolah menjadi sinyal output PWM. Tentunya nilai yang dibandingkan pada arduino bukanlah nilai PWM sebagai mana dijelaskan system kerja drive ESC pada bagian sebelumnya. Namun, nilai PWM yang terbangkit dari sinyal analog yang dikirimkan oleh potensiometer diubah kedalam bentuk referensi rpm dengan cara menskala hasil rpm yang dikirimkan kedalam ESC dan juga kecepatan yang terbangkit. Karena setiap kenaikan nilai PWM menyebabkan nilai putaran bertambah dengan rentan nilai yang mendekati sama. Maka nilai itu kemudian diskala dan dibuat sebagai nilai referensi masukan ke dalam ESC.

Adapun nilai PWM yang masuk ke dalam ESC akan selalu berubah-ubah dan tidak sama seperti pada kontrol ESC sebelumnya. Hal ini diakibatkan oleh eror pembacaan sensor optocoupler yang digunakan. Dalam kontrol steering motor ini digunakan salah satu motor sebagai referensinya dan motor yang lain akan mengikutinya.

Nilai kecepatan motor akan sama apabila nilai yang referensi dari arduino juga memiliki nilai yang sama.

Berikut merupakan blok diagram sistem kontrol pada motor:



Gambar 3. 3. Blok Diagram Sistem Kontrol

Pada kondisi berbelok, kontrol ackerman-jeantand berlaku dimana saat sudut positif atau negative akan mengakibatkan penambahan atau pengurangan nilai PWM. Hal ini juga sudah diperhitungkan agar nilai yang dihasilkan sesuai. Adapun flowchart fuzzy control pada penelitian ini terlihat pada gambar 3.3:

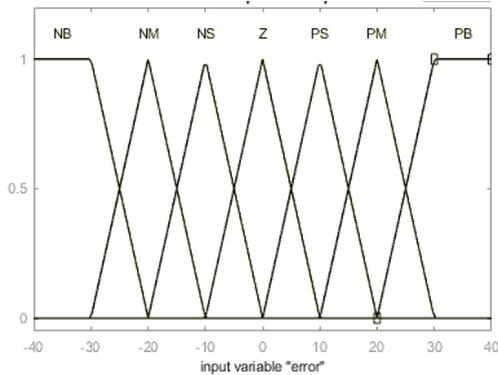
Pada sistem kontrol kecepatan pada fuzzy logic control yang digunakan pada penelitian ini mempunyai nilai referensi berasal dari nilai potensiometer yang awalnya bernilai analog dari 0 – 1023 menjadi sebuah sinyal digital dari 0 – 180. Nilai potensiometer ini kemudian dikalkulasikan dengan cara menskalakan PWM input pada ESC dan kecepatan actual yang ditimbulkan oleh motor. Kecepatan referensi ini kemudian diubah dalam bentuk PWM sebelum masuk kedalam esc, hal ini dikarenakan untuk dapat memutar motor maka nilai yang diinputkan kedalam ESC harus mulai dari nilai yang kecil.

Pada PWM ini akan menyebabkan motor berputar yang kemudian putaran motor akan dibaca sensor kecepatan. Kecepatan actual ini akan dibandingkan dengan kecepatan referensi di dalam kontrol fuzzy. Pada FLC terdapat beberapa proses. Pertama yaitu fuzzyfikasi atau pengenalan eror dan delta error. Error ini adalah perbandingan antara kecepatan actual dan referensi. Di dalam kontrol fuzzy telah ditetapkan rule-base atau suatu kondisi yang akan diakibatkan dari adanya proses fuzzyfikasi.

3.4.1. Proses Fuzzifikasi

Diagram *crisp* yang digunakan dalam proses fuzzifikasi pada penelitian ini dirancang dengan menggunakan fungsi keanggotaan segitiga dan setengah trapesium pada masing-masing tepi sisi kanan

dan kirinya. Rentang nilai fuzzifikasi akan ditetapkan mulai dari -40 sampai dengan 40. Sebelum dilakukan proses fuzzifikasi akan dilakukan proses normalisasi pada sinyal error dan delta-error sesuai dengan rentang nilai fuzzifikasi mulai dari -40 sampai dengan 40 tersebut. Hal ini didasarkan pada keadaan trend error sebelum ditambahkan sistem kontrol.



Gambar 3.4. membership function sinyal error

3.4.2. Penentuan *Rule Base*

Setelah melakukan proses fuzzifikasi langkah selanjutnya menentukan *rule-base* (aturan) terhadap masing-masing fungsi keanggotaan dari nilai error *controller* tersebut. Perancangan *rule-base* menyesuaikan dengan jumlah fungsi keanggotaan untuk proses fuzzifikasi dan defuzzifikasi menggunakan metode *Mamdani*. Pada *Controller*, perancangan *rule* ditetapkan dengan tujuh buah *reference* yang diantaranya adalah *Very Negative* (VN), *Medium Negative* (MN), *Small Negative* (SN), *Zero Equals* (ZE), *Small Positive* (SP), *Medium Positive* (MP), serta *Very Positive* (VP). Masing-masing hubungan fungsi keanggotaan dapat dinyatakan dalam tabel 3.2.

Tabel 3.2. Rule Base sistem kontrol

e	NB	NM	NS	ZE	PS	PM	PB
Δe							
ZE	VN	MN	SN	ZE	SP	MP	VP

Dalam penentuan *rule-base* hanya didasarkan pada nilai error *controller* sedangkan pada nilai delta error tidak disertakan, karena

akan mengakibatkan nilai eror semakin besar. Hal ini disebabkan oleh delay pada program kontrol sehingga nilai delta eror tidak real time.

3.4.3. Inferensi Mamdani

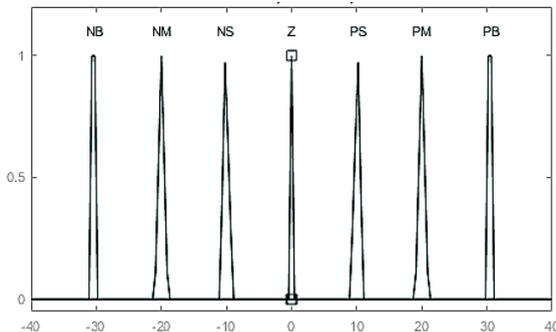
Langkah selanjutnya adalah melakukan proses inferensi, dimana dalam dalam pengaturan level ini, metode inferensi yang digunakan adalah *minimum of maksimum* (MOM), yang dinyatakan dalam persamaan (3.1).

$$\mu_x = \max(\mu_x, \min(mf(i))) \dots (3.1)$$

Variabel μ_x menyatakan hasil inferensi, $mf1$ merupakan nilai dari masing-masing fungsi keanggotaan untuk $i = 1, 2, \dots$ (jumlah fungsi keanggotaan nilai eror).

3.4.4. Proses Defuzzifikasi

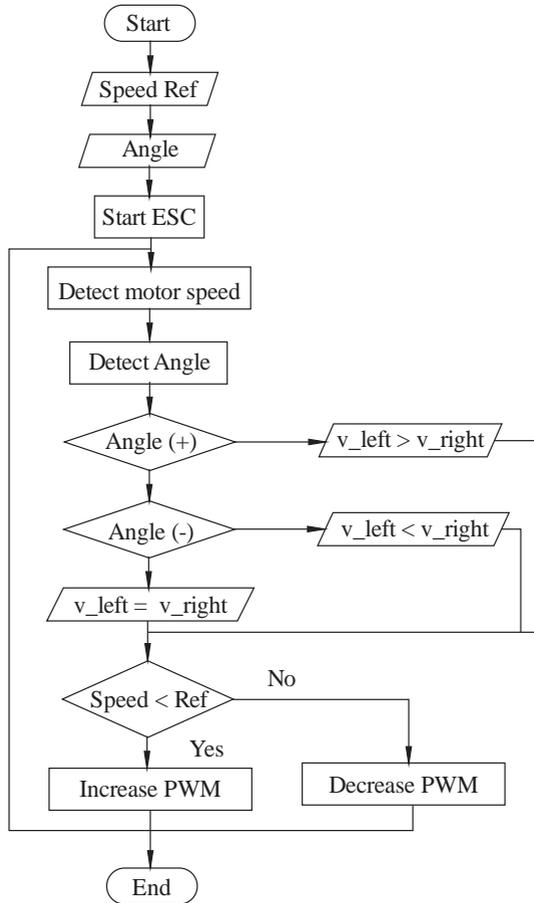
Pada tahap ini nilai-nilai dari *reference* yang ditentukan pada *rulebase* akan dinyatakan dalam diagram *crisp*. Adapun nilai pada proses defuzzifikasi sistem *Controller* fungsi keanggotaan menggunakan tipe *singleton* yang direpresentasikan ke dalam diagram *crisp* seperti pada *Gambar 3.5*.



Gambar 3.5. Defuzzifikasi *singleton* pada sistem kontrol

Pada *Controller* dirancang nilai sinyal kontrol u_x yang simetri, yaitu ditetapkan pada nilai $\{-30, -20, -10, 0, 10, 20, 30\}$. Hal ini didasarkan nilai pengendalian sudut pada sistem kontroler.

Adapun diagram alir pembuatan sistem kontrol pada penelitian ini ditunjukkan gambar 3.6.



Gambar 3. 6. Skema Flowchart Sistem Kontrol

--- halaman ini sengaja dikosongkan ---

BAB 4

HASIL IMPLEMENTASI DAN PEMBAHASAN

Pengujian dilakukan dengan cara membuat implementasi dari simulasi pada penelitian sebelumnya yang telah dilakukan oleh mahasiswa konversi energi. Dengan menggunakan parameter yang sama, penulis melakukan beberapa modifikasi dalam menyederhanakan proses guna memudahkan dalam pembuatan alat. Maka diperoleh hasil sebagai berikut :

4.1. Pengujian sensor kecepatan optocoupler

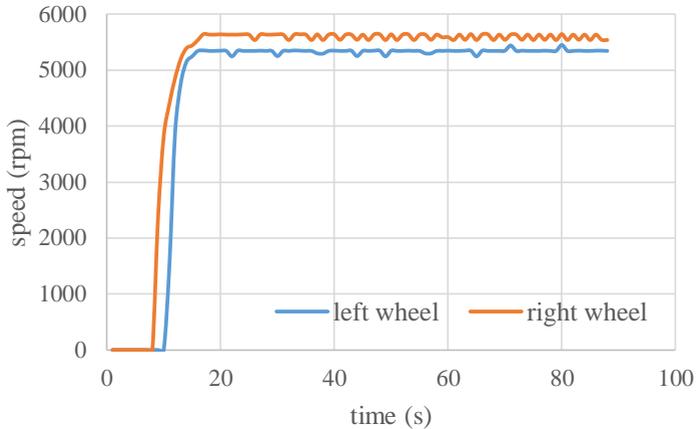
Pada implementasi alat pengendali kecepatan, penulis menggunakan sensor optocoupler untuk membaca kecepatan putaran pada motor. Pengujian sensor ini dilakukan dengan membandingkan hasil perhitungan sensor dan tachometer TM-300. Pengujian dilakukan menggunakan sumber baterai lipo 3 sel dengan tegangan 12V. Pada ESC diprogram menggunakan duty cycle 50% (diambil dari library servo) dan PWM berfungsi sebagai pengaturan amplitudo. Didapatkan hasil pengujian sebagai berikut :

Tabel 4.1 Pengujian Sensor Kecepatan Optocoupler

PWM	Pembacaan Kecepatan		Error pembacaan %
	Optocoupler (rpm)	TM-300 (rpm)	
13	3061	3047	0,45
15	3715	3723	0,21
14	3426	3437	0,32
16	3789	3792	0,08
17	4122	4133	0,27
20	4661	4679	0,38
25	5433	5445	0,22
30	6064	6084	0,33
Rata-rata error pembacaan			0,28

4.2. Pengujian tes PWM sebelum kalibrasi

Pada pengujian ini dilakukan dengan memberikan perintah arduino untuk mengirimkan sinyal PWM dengan nilai yang sama kepada kedua ESC. Pengujian ini bertujuan untuk mengetahui respon kecepatan pada roda kanan dan roda kiri sebelum dilakukan proses kalibrasi pada kedua ESC. berikut adalah hasil uji :

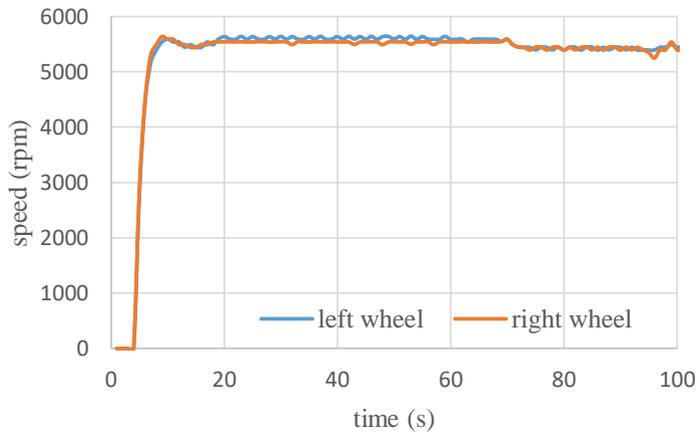


Gambar 4.1. Grafik kecepatan sebelum kalibrasi

Pada saat sebelum dilakukan kalibrasi, menunjukkan data bahwa kecepatan kedua roda berbeda meskipun diberikan sinyal input yang sama. Seperti yang telah dibahas pada bab sebelumnya bahwa nilai ESC antara yang satu dengan yang lain belum tentu sama pada kondisi awalnya. Oleh karena itu perlu adanya proses kalibrasi.

4.3. Pengujian tes PWM setelah kalibrasi

Setelah dilakukan kalibrasi terdapat perubahan nilai pada kedua ESC. dimana dengan sinyal input yang sama mengakibatkan output putaran yang mendekati sama. Namun masih terdapat eror yang cukup besar dikarenakan belum adanya kontrol pada kedua ESC. berikut adalah grafik kecepatan setelah kalibrasi.



Gambar 4.2. Grafik kecepatan setelah kalibrasi

4.4. Pengujian steering motor kontrol fuzzy

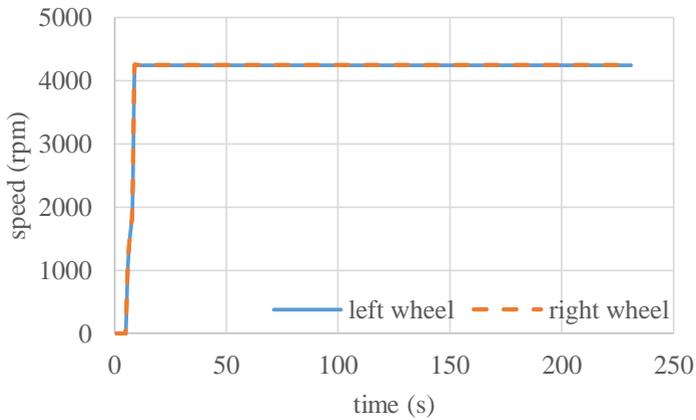
4.4.1. Sudut $\delta = 0$ (pada kondisi jalan lurus)

Pada pembuatan implementasi alat, penulis membatasi kecepatan referensi pada motor sebesar 100 - 130 m/s atau kurang dari 13000 rpm. Hal ini dikarenakan keterbatasan pembacaan sensor kecepatan yang cenderung terjadi error pada rpm rendah dan juga terlalu tinggi.

Pada pengujian sudut $\delta = 0$, kecepatan rpm diset pada 444.4 rad/s. Atau untuk mengubah ke dalam bentuk rpm maka,

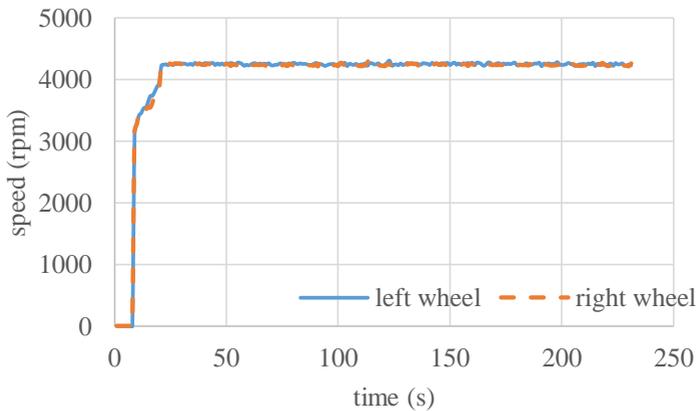
$$n = \frac{\omega \cdot 60}{2 \cdot \pi} = 4252 \text{ rpm.}$$

Dikarenakan sudut $\delta = 0$, maka $n_{\text{roda kanan}} = n_{\text{roda kiri}}$. Berikut adalah grafik hasil uji :



Gambar 4.3. Referensi kecepatan dengan sudut 0

Terlihat dari grafik bahwa referensi untuk roda kanan dan kiri adalah sama. Namun pada realnya, terdapat sedikit perbedaan nilai antara referensi kecepatan putar roda kanan dan referensi kecepatan putar kiri. Hal ini disebabkan oleh nilai dari potensiometer yang tidak bisa memberikan hasil yang benar-benar tepat meskipun sudah diatur. Berikut adalah hasil pembacaan sensor kecepatan roda kanan dan roda kiri atau bisa disebut dengan kecepatan aktual.

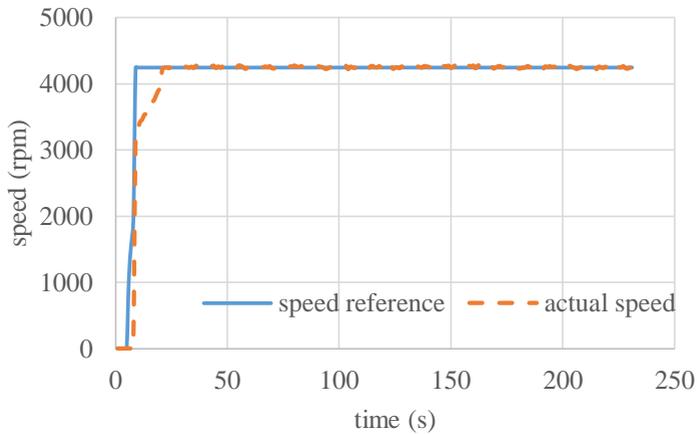


Gambar 4.4. Kecepatan aktual dengan sudut 0

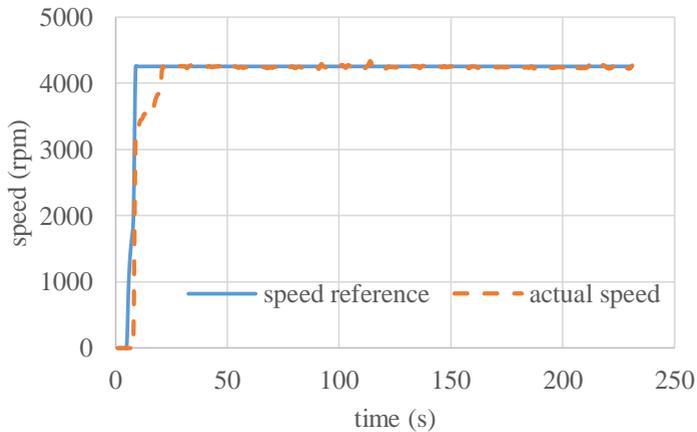
Dari data tabel, berikut merupakan grafik perbandingan antara kecepatan referensi dan kecepatan aktual roda kiri maupun roda kanan. Dari grafik terlihat bahwa kecepatan putar yang terbaca pada sensor kecepatan terdapat error dan hal ini mengakibatkan kecepatan actual dan kecepatan referensi tidak sepenuhnya sama. Error rata-rata pada pengujian dengan sudut $\delta = 0$ menunjukkan angka 0,27% untuk roda kiri dan 0,167% untuk roda kanan.

Tabel 4.2 Error Pembacaan pada sudut 0

Referensi roda kiri (rpm)	Aktual roda kiri (rpm)	Error (rpm)	Referensi roda kanan (rpm)	Aktual roda kanan (rpm)	Error (rpm)
4243,5	4233	10,5	4252	4247	5
4243,5	4248	-4,5	4252	4263	-11
4243,5	4268	-24,5	4252	4243	9
4243,5	4248	-4,5	4252	4261	-9
4243,5	4264	-20,5	4252	4256	-4
4243,5	4257	-13,5	4252	4251	1
4243,5	4258	-14,5	4252	4253	-1
4243,5	4248	-4,5	4252	4225	27
4243,5	4240	3,5	4252	4255	-3
4243,5	4264	-20,5	4252	4257	-5
4243,5	4257	-13,5	4252	4253	-1
4243,5	4253	-9,5	4252	4266	-14
4243,5	4235	8,5	4252	4250	2
Rata-rata error		11,73	Rata-rata error		7,07



Gambar 4.5. Aktual dan referensi roda kiri dengan sudut 0



Gambar 4.6. Aktual dan referensi roda kanan dengan sudut 0

4.4.2. Sudut $\delta = 15$ (pada kondisi belok kanan)

Pada pengujian sudut $\delta = 15$, penulis menggunakan set referensi kecepatan adalah 157,8 rad/s.

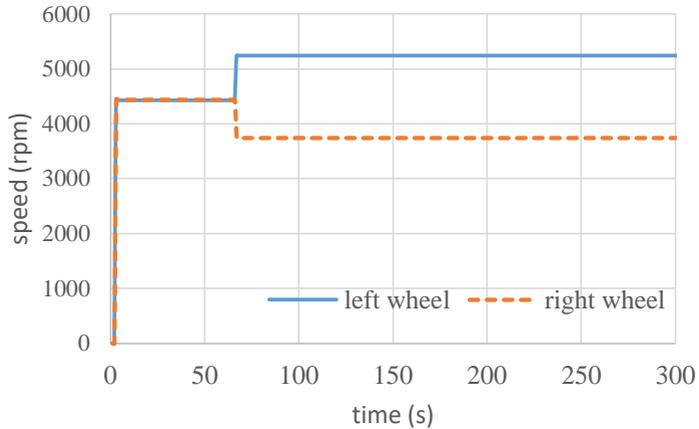
Untuk mengubah nilai menjadi rpm maka nilai tersebut menjadi,

$$n = \frac{\omega \cdot 60}{2\pi} = 1506 \text{ rpm.}$$

Pada saat sebelum terjadi perubahan sudut, kecepatan kedua roda adalah sama yaitu sebesar, 463.9 rad/s.

Untuk mengubah ke dalam bentuk rpm maka,

$$n = \frac{\omega \cdot 60}{2\pi} = 4430 \text{ rpm.}$$

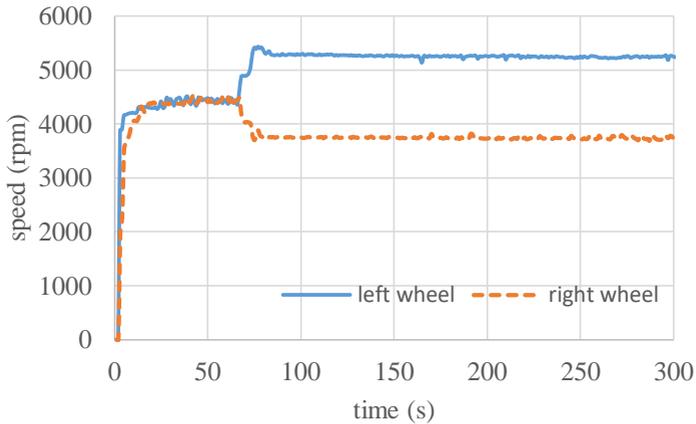


Gambar 4.7. Referensi kecepatan dengan sudut 15

Terlihat dari grafik bahwa referensi untuk roda kanan dan kiri adalah sama sesaat sebelum terjadinya perubahan sudut. Namun pada realnya, terdapat sedikit perbedaan nilai antara referensi kecepatan putar roda kanan dan referensi kecepatan putar kiri. Hal ini disebabkan oleh nilai dari potensiometer yang tidak bisa memberikan hasil yang benar-benar tepat meskipun sudah diatur.

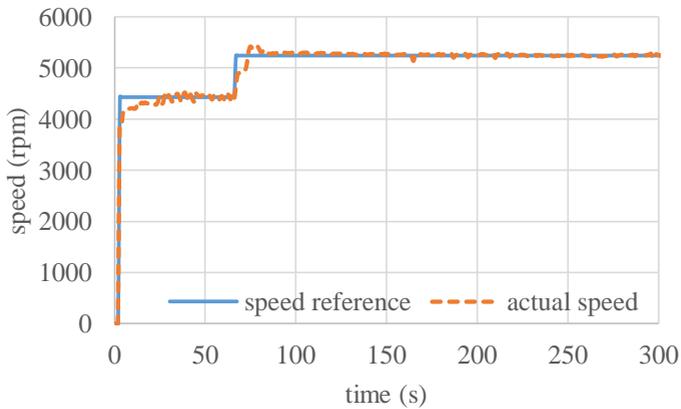
Berikut adalah hasil pembacaan sensor kecepatan roda kanan dan roda kiri atau bisa disebut dengan kecepatan aktual. Saat setelah terjadi perubahan sudut sebesar 15 derajat, maka pada kecepatan putar referensi roda kiri dan roda kanan juga mengalami perubahan sebesar : $n_l = \omega_{ref} + \Delta\omega/2$ dan $n_r = \omega_{ref} - \Delta\omega/2$. Sehingga kecepatan referensi putar roda kiri menjadi 5183 dan kecepatan referensi roda kanan sebesar 3667. Namun pada aktualnya terjadi sedikit perbedaan nilai, hal ini disebabkan oleh kalkulasi nilai potensiometer yang tidak bisa menunjukkan nilai yang tepat. Pada nilai real, besarnya referensi kecepatan putar roda kiri adalah sebesar

5242 rpm dan roda kanan sebesar 3741 rpm. Berikut adalah grafik kecepatan aktual roda kiri dan kanan.



Gambar 4.8. Kecepatan aktual dengan sudut 15

Pada kecepatan aktual, terjadi error yang dibandingkan dengan kecepatan referensi. Data ini diambil setelah kecepatan putar pada motor dalam keadaan sudah steady state. Diambil beberapa sampel data dan tertera pada tabel 4.3 :

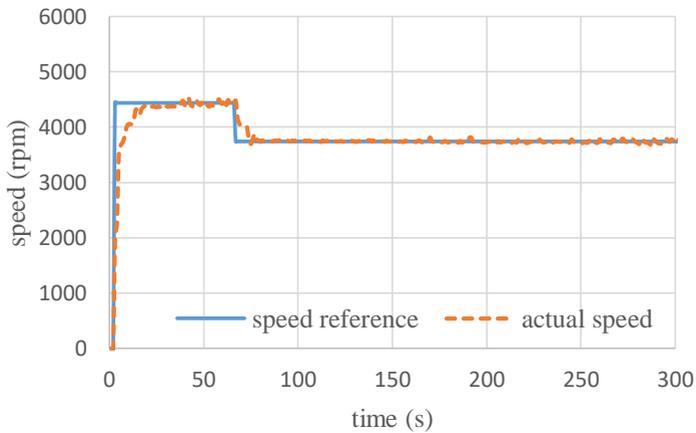


Gambar 4.9. Aktual dan referensi roda kiri dengan sudut 15

Tabel 4.3 Error Pembacaan pada sudut 15

Referensi roda kiri (rpm)	Aktual roda kiri (rpm)	Error (rpm)	Referensi roda kanan (rpm)	Aktual roda kanan (rpm)	Error (rpm)
5242,92	5238	4,92	3741,09	3715	26,09
5242,92	5231	11,92	3741,09	3771	-29,91
5242,92	5260	-17,08	3741,09	3747	-5,91
5242,92	5242	0,92	3741,09	3724	17,09
5242,92	5241	1,92	3741,09	3729	12,09
5242,92	5230	12,92	3741,09	3736	5,09
5242,92	5250	-7,08	3741,09	3715	26,09
5242,92	5245	-2,08	3741,09	3720	21,09
5242,92	5235	7,92	3741,09	3739	2,09
5242,92	5256	-13,08	3741,09	3739	2,09
5242,92	5250	-7,08	3741,09	3736	5,09
5242,92	5238	4,92	3741,09	3736	5,09
5242,92	5242	0,92	3741,09	3724	17,09
Rata-rata error		7,13	Rata-rata error		13,44

Dari data sampel diatas, berikut merupakan grafik perbandingan antara kecepatan referensi dan kecepatan aktual roda kiri maupun roda kanan. Dari grafik terlihat bahwa kecepatan putar yang terbaca pada sensor kecepatan terdapat error dan hal ini mengakibatkan kecepatan actual dan kecepatan referensi tidak sepenuhnya sama. Error rata-rata pada pengujian dengan sudut $\delta = 15$ menunjukkan angka 0,14% untuk roda kiri dan 0,36% untuk roda kanan.



Gambar 4.10. Aktual dan referensi roda kanan dengan sudut 15

4.4.3. Sudut $\delta = -25$ (pada kondisi belok kiri)

Pada pengujian sudut $\delta = -25$, penulis menggunakan kecepatan referensi perbedaan kecepatan antara roda kanan dan roda kiri sebesar 302.9 rad/s.

Untuk mengubah nilai menjadi rpm maka nilai tersebut menjadi,

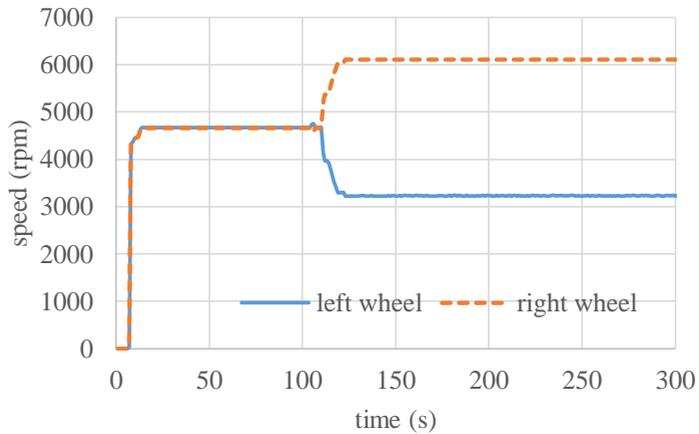
$$n = \frac{\omega \cdot 60}{2\pi} = 2890 \text{ rpm.}$$

Pada saat sebelum terjadi perubahan sudut, referensi kecepatan antara roda kanan dan roda kiri adalah sama yaitu sebesar, 489 rad/s.

Untuk mengubah ke dalam bentuk rpm maka,

$$n = \frac{\omega \cdot 60}{2\pi} = 4670 \text{ rpm.}$$

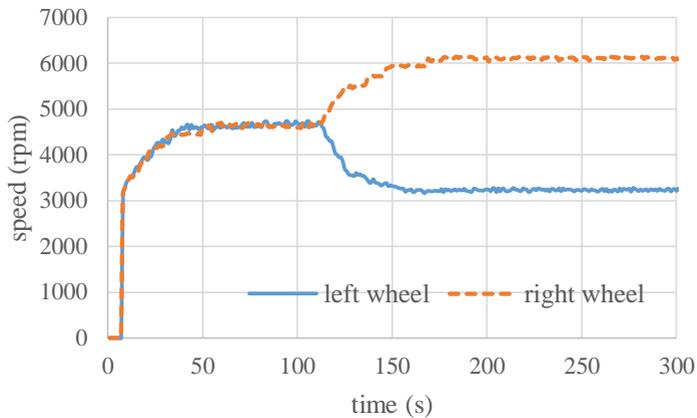
Terlihat dari grafik 4.11. bahwa referensi untuk roda kanan dan kiri adalah sama sesaat sebelum terjadinya perubahan sudut. Namun pada realnya, terdapat sedikit perbedaan nilai antara referensi kecepatan putar roda kanan dan referensi kecepatan putar kiri. Hal ini disebabkan oleh nilai dari potensiometer yang tidak bisa memberikan hasil yang benar-benar tepat meskipun sudah diatur.



Gambar 4.11. Referensi kecepatan dengan sudut -25

Berikut adalah hasil pembacaan sensor kecepatan roda kanan dan roda kiri atau bisa disebut dengan kecepatan aktual. Saat setelah terjadi perubahan sudut sebesar -25 derajat, maka pada kecepatan putar referensi roda kiri dan roda kanan juga mengalami perubahan sebesar : $n_l = \omega_{ref} + \Delta\omega/2$ dan $n_r = \omega_{ref} - \Delta\omega/2$. Sehingga kecepatan referensi putar roda kiri menjadi 3225 dan kecepatan referensi roda kanan sebesar 6115. Namun pada aktualnya terjadi sedikit perbedaan nilai, hal ini disebabkan oleh kalkulasi nilai potensiometer yang tidak bisa menunjukkan nilai yang tepat. Pada nilai real, besarnya referensi kecepatan putar roda kiri adalah sebesar 3224 rpm dan roda kanan sebesar 6111 rpm. Berikut adalah grafik kecepatan aktual roda kiri dan kanan.

Pada kecepatan aktual, terjadi error yang dibandingkan dengan kecepatan referensi. Data ini diambil setelah kecepatan putar pada motor dalam keadaan sudah steady state. Diambil beberapa sampel data dan berikut merupakan hasil uji kecepatan



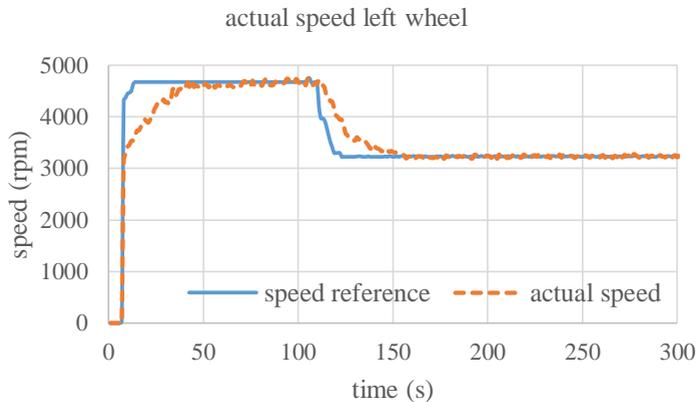
Gambar 4.12. Kecepatan aktual dengan sudut -25

Tabel 4.4 Error Pembacaan pada sudut -25

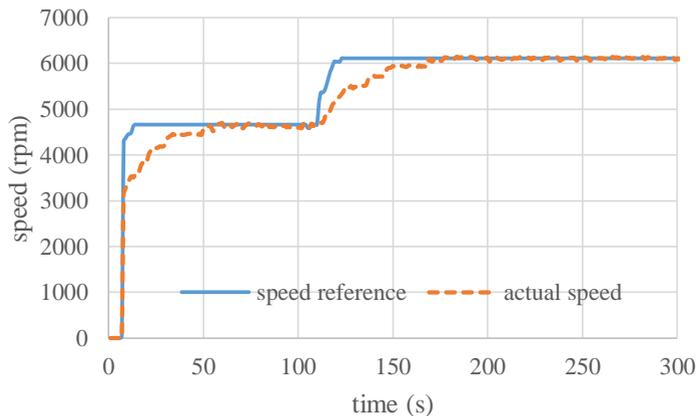
Referensi roda kiri (rpm)	Aktual roda kiri (rpm)	Error (rpm)	Referensi roda kanan (rpm)	Aktual roda kanan (rpm)	Error (rpm)
3230,58	3245	-14,42	6111,63	6129	-4,94
3239,94	3225	14,94	6111,63	6128	-13,31
3230,58	3247	-16,42	6111,63	6101	23,06
3233,37	3229	4,37	6111,63	6136	-33,71
3224,02	3225	-0,98	6111,63	6128	-16,37
3224,02	3200	24,02	6111,63	6112	-0,37
3239,94	3204	35,94	6111,63	6123	-8,31
3230,58	3240	-9,42	6111,63	6112	12,06
3230,58	3257	-26,42	6111,63	6117	7,06
3224,02	3207	17,02	6111,63	6094	17,63
3230,58	3255	-24,42	6111,63	6149	-24,94
3224,02	3193	31,02	6111,63	6103	8,63
3233,37	3223	10,37	6111,63	6124	-21,71
Rata-rata error		17,67	Rata-rata error		14,77

Dari data sampel diatas, berikut merupakan grafik perbandingan antara kecepatan referensi dan kecepatan aktual roda

kiri maupun roda kanan. Dari grafik terlihat bahwa kecepatan putar yang terbaca pada sensor kecepatan terdapat error dan hal ini mengakibatkan kecepatan actual dan kecepatan referensi tidak sepenuhnya sama. Error rata-rata pada pengujian dengan sudut $\delta = -25$ menunjukkan angka 0,14% untuk roda kiri dan 0,55% untuk roda kanan.



Gambar 4.13. Aktual dan referensi roda kiri dengan sudut -25



Gambar 4.14. Aktual dan referensi roda kanan dengan sudut -25

--- halaman ini sengaja dikosongkan ---

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Pada penelitian ini, terdapat beberapa kesimpulan diantaranya adalah :

1. Metode pengaturan Ackermann-jeantand dapat diimplementasikan untuk pengaturan kecepatan penggerak roda dengan terpisah
2. Implementasi Ackermann-Jeantand dengan open loop mengalami eror sebesar 3-5%
3. Penambahan close-loop pada sistem (fuzzy logic controller) pengaturan kecepatan mengakibatkan eror menjadi 0.2-1%

5.2. Saran

1. Penggunaan *Brushless* DC sensored lebih di rekomendasikan karena memiliki sensor kecepatan yang lebih akurat dibandingkan dengan sensor optocoupler. Hal ini akan mengakibatkan hasil yang didapatkan lebih stabil.
2. Perancangan kontroler fuzzy sebaiknya ditambahkan nilai rule-basenya sehingga hasil output yang dihasilkan bisa lebih sempurna.

--- halaman ini sengaja dikosongkan ---

DAFTAR PUSTAKA

- [1] C. Chhith, "Modelling and Simulation of Independent Speed Steering Control for Front In-Wheel in EV Using BLDC Motor in MATLAB GUI", Indonesia, hal 1-7, 2018
- [2] J. N. Ansari dan S. L, "Speed Control of BLDC Motor for Electric Vehicle," *International Journal of Engineering Research & Technology (IJERT)*, vol. 3, pp. 1666 - 1671, 2014.
- [3] J. Zhao dan Y. Yu. Brushless DC Motor Fundamentals, 2011.
- [4] L. Yen-Shin, S. Fu-San, dan C. Yung-Hsin, "Novel Loss Reduction Pulsewidth Modulation Technique for Bushless DC Motor Drives Fed by MOSFET Inverter," *IEEE Transactions on Power Electronics*, vol. 19, pp. 1646-1652, 2004.
- [5] R. Arulmozhiyal dan R. Kandiban, "Design of Fuzzy PID Controller for Brushless DC Motor," in *2012 International Conference on Computer Communication and Informatics*, pp. 1-7, 2012.
- [6] K. Premkumar dan B. V. Manikandan, "Adaptive Neuro-Fuzzy Inference System based Speed Controller for Brushless DC Motor," *ELSEVIER Neurocomputing*, vol. 138, pp. 260-270, 2014.
- [7] C.-L. Xia, *Permanent Magnet Brushless DC Motor Drives and Controls*: John Wiley & Sons Singapore Pte. Ltd., 2012.
- [8] P. Yedamale dan M. T. Inc. Brushless DC (BLDC) Motor Fundamentals. 20, 2003.
- [9] H. Mohan, R. K. P, dan G. S, "Speed Control of Brushless DC Motor Using Fuzzy Based Controllers," *IRJET*, vol. 2, pp. 875-881, 2015.
- [10] Hartono, "Optimization of Tsukamoto Fuzzy Inference System using Fuzzy Grid Partition," *International Journal of Computer Science and Network*, vol. 5, p. 6, 5 October 2016 2016.

-- halaman ini sengaja dikosongkan ---

LAMPIRAN

A. Program Kalibrasi ESC

```
#include <Servo.h> //library motor servo

Servo ESC1; //membuat objek servo untuk kontrol ESC

Servo ESC2; //membuat objek servo untuk kontrol ESC

int motor_speed;

uint8_t flag = 0; //inisialisasi pengunci nilai

void setup() {

    //melampirkan ESC pada pin 9

    ESC1.attach(9,1000,2000); // (pin, lebar pulsa minimal, lebar pulsa
maksimal dalam sekon)

    ESC2.attach(10,1000,2000); //(pin, lebar pulsa minimal, lebar pulsa
maksimal dalam sekon)

    Serial.begin(38400);

    pinMode(A0, INPUT);

    pinMode(A1, INPUT);

    pinMode(7, INPUT_PULLUP);}

void loop() {

    if(digitalRead(7)){

        flag == 0;

        ESC1.write(180); // mengirimkan sinyal ke ESC

        ESC2.write(180); // mengirimkan sinyal ke ESC

    }

}
```

```

else if(!digitalRead(7) && flag == 0){
    flag = 1;
    Serial.println("wait");
    ESC1.write(0);
    ESC2.write(0);
    delay(10000);
    Serial.println("test");
    ESC1.write(20);
    ESC2.write(20);
    delay(500);
    ESC1.write(0);
    ESC2.write(0);
    delay(1000);
    ESC1.write(20);
    ESC2.write(20);
    delay(500);
    ESC1.write(0);
    ESC2.write(0);
    delay(1000);
    Serial.println("end");}
else if(flag == 1){
    motor_speed = analogRead(A1); //masukan dari sinyal analog
    motor_speed = map(motor_speed, 0, 1023, 0, 180);

```

```
Serial.print(motor_speed);  
Serial.print(" ");  
Serial.println((int)(motor_speed));  
  
ESC1.write((int)(motor_speed)); // mengirimkan sinyal ke ESC  
ESC2.write((int)(motor_speed)); // mengirimkan sinyal ESC  
}  
}
```

-- halaman ini sengaja dikosongkan ---

B. Program Open Loop

```
#include <Servo.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);
Servo ESC1; // create servo object to control the ESC
Servo ESC2;

float potValue_1 = 0, potValue_2 = 0;;
float motor_speed;
float multiplier_1 = 0, multiplier_2 = 0;
float valueL=0, valueR=0;
float revL=0, revR=0;
float rpmL=0, rpmR=0;
float oldtimeL =0, oldtimeR=0;
float timeL, timeR;
float a1=0;
float a2=0;
float rpmRsampling=0;
float rpmLsampling=0;
float rpmRcount=0;
float rpmLcount=0;
unsigned long previousMillis1=0;
unsigned long previousMillis2=0;
```

```

long interval = 5000;

void isrL(){ //interrupt service routine
revL++;}

void isrR(){ //interrupt service routine
revR++;}

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

ESC1.attach(9,1000,2000); // (pin, min pulse width, max pulse width
in milliseconds)

ESC2.attach(10,1000,2000); // (pin, min pulse width, max pulse
width in milliseconds)

pinMode(A0, INPUT);
pinMode(A1, INPUT);

attachInterrupt(0,isrL,RISING); //attaching the interrupt
attachInterrupt(1,isrR,RISING);}

void speedL(){
detachInterrupt(1); //detaches the interrupt
timeL = millis()-oldtimeL; //finds the time
rpmL =(revL/timeL)*60000; //calculates rpm
oldtimeL=millis(); //saves the current time
revL=0;

attachInterrupt(1,isrL,RISING);}

```

```

void speedR(){
detachInterrupt(0);      //detaches the interrupt
timeR = millis()-oldtimeR;    //finds the time
rpmR=(revR/timeR)*60000;    //calculates rpm
oldtimeR=millis();        //saves the current time
revR=0;

attachInterrupt(0,isrR,RISING);}

void loop() {

potValue_1 = analogRead(A0); // reads the value of the
potentiometer (value between 0 and 1023)

potValue_2 = 1023 - potValue_1;

multiplier_1 = map(potValue_2, 0, 1023, 0, 1000); // scale it to use
it with the servo library (value between 0 and 180)

multiplier_2 = map(potValue_1, 0, 1023, 0, 1000); // scale it to use
it with the servo library (value between 0 and 180)

motor_speed = analogRead(A1);

motor_speed = map(motor_speed, 0, 1023, 0, 180);

speedL();

speedR();

Serial.print("\t reference = ");

Serial.print(motor_speed);

Serial.print("\t L = ");

Serial.print((float)(motor_speed*multiplier_1/1000));

Serial.print("\t R = ");

Serial.print((float)(motor_speed*multiplier_2/1000));

```

```

Serial.print("  RPML= ");
Serial.print(rpmL);
Serial.print("  RPMR= ");
Serial.println(rpmR);
/*
unsigned long currentMillis2 = millis();
if (currentMillis2 - previousMillis2 < interval) {
rpmLcount=rpmLcount+rpmL;
a1 = a1 + 1;
rpmLsampling = rpmLcount / a1 ;}
else {
Serial.print("  RPMLsampling= ");
Serial.println(rpmLsampling);
rpmLcount = 0.0;
a1 = 0;
previousMillis2 = currentMillis2;}
unsigned long currentMillis1 = millis();
if (currentMillis1 - previousMillis1 < interval) {
rpmRcount=rpmRcount+rpmR;
a2= a2 + 1;
rpmRsampling = rpmRcount / a2 ;}
else {
Serial.print("  RPMRsampling= ");

```

```

    Serial.println(rpmRsampling);
    rpmRcount = 0.0;
    a2 = 0;
    previousMillis1 = currentMillis1;}
*/

lcd.init();
lcd.backlight();
lcd.setCursor(0,0);
lcd.print("RPML = ");
lcd.print(rpmL);
lcd.setCursor(0,1);
lcd.print("RPMR = ");
lcd.print(rpmR);
ESC1.write((float)(motor_speed*multiplier_1/1000)); // Send the
signal to the ESC
ESC2.write((float)(motor_speed*multiplier_2/1000));
delay(100);}

```

--- halaman ini sengaja dikosongkan ---

C. Program Close-Loop

```
#include <Servo.h>
```

```
#include <Wire.h>
```

```
#include <LiquidCrystal_I2C.h>
```

```
LiquidCrystal_I2C lcd(0x27,16,2);
```

```
Servo ESC1; // create servo object to control the ESC
```

```
Servo ESC2;
```

```
float potValue_1 = 0, potValue_2 = 0;;
```

```
float motor_speed;
```

```
float multiplier_1 = 0, multiplier_2 = 0;
```

```
float valueL=0, valueR=0;
```

```
float revL=0, revR=0;
```

```
int rpmL, rpmR;
```

```
int oldtimeL =0, oldtimeR=0;
```

```
int timeL, timeR;
```

```
int rule[7] = {0,1,2,3,4,5,6};
```

```
int n,i=0;
```

```
float k,l=0;
```

```
unsigned long v=0;
```

```
float c[7] = {-30,-20,-10,0,10,20,30};
```

```
float en,den,d=0;
```

```

float x1,x2,a,b,z1,z2=0;
float j1,j2=0;

float fuzzy1(){
float mf[7]={0,0,0,0,0,0,0};
float u[7]={0,0,0,0,0,0,0};
//Serial.println(i);
en=k*0.00015;
//Serial.println(en);
if(en<=-3) mf[0]=1;
else if(en<=-2){
    mf[0]=-2-en;
    mf[1]=en+3;}
else if(en<=-1){
    mf[1]=-1-en;
    mf[2]=en+2;}
else if(en<=0){
    mf[2]=-en;
    mf[3]=en+1;}
else if(en<=1){
    mf[3]=1-en;
    mf[4]=en;}
else if(en<=2){

```

```

mf[4]=2-en;
mf[5]=en-1;}
else if(en<=3) {
mf[5]=3-en;
mf[6]=en-2;}
else mf[6]=1;

for (i=0;i<7;i++){
n = rule[i];
a=mf[i];
b=u[n];
u[n]=max(b,a);}
a=0; b=0;
for(i=0;i<7;i++){
a=a+u[i]*c[i];
b=b+u[i];}
b=b*180;
x1=a/b;
x1=x1*180;
j1=j1+0.07*x1; //ki
z1=0.2*x1+j1; //kp
if(z1<9&&rpmL) {z1=9; j1=9;}
if(motor_speed*multiplier_1/1000<100) {z1=0;j1=0;}

```

```

//Serial.println(z1);
return z1;}

float fuzzy2(){
float mf[7]={0,0,0,0,0,0,0};
float u[7]={0,0,0,0,0,0,0};
//Serial.println(i);
en=1*0.00015;
//Serial.println(en);
if(en<=-3) mf[0]=1;
else if(en<=-2){
    mf[0]=-2-en;
    mf[1]=en+3;}
else if(en<=-1){
    mf[1]=-1-en;
    mf[2]=en+2;}
else if(en<=0){
    mf[2]=-en;
    mf[3]=en+1;}
else if(en<=1){
    mf[3]=1-en;
    mf[4]=en;}
else if(en<=2){

```

```

mf[4]=2-en;
mf[5]=en-1;}
else if(en<=3) {
mf[5]=3-en;
mf[6]=en-2;}
else mf[6]=1;

for (i=0;i<7;i++){
n = rule[i];
a=mf[i];
b=u[n];
u[n]=max(b,a);}
a=0; b=0;
for(i=0;i<7;i++){
a=a+u[i]*c[i];
b=b+u[i];}
b=b*180;
x2=a/b;
x2=x2*180;
j2=j2+0.07*x2; //ki
z2=0.2*x2+j2; //kp
if(z2<9&&rpmR) {z2=9; j2=9;}
if(motor_speed*multiplier_2/1000<100) {z2=0;j2=0;}

```

```
//Serial.println(z2);
```

```
return z2;}
```

```
void isrL(){ //interrupt service routine
```

```
revL++;}
```

```
void isrR(){ //interrupt service routine
```

```
revR++;}
```

```
void setup() {
```

```
  Serial.begin(9600);
```

```
  ESC1.attach(9,1000,2000); // (pin, min pulse width, max pulse  
width in milliseconds)
```

```
  ESC2.attach(10,1000,2000); // (pin, min pulse width, max pulse  
width in milliseconds)
```

```
  pinMode(A0, INPUT);
```

```
  pinMode(A1, INPUT);
```

```
  attachInterrupt(0,isrL,RISING); //attaching the interrupt
```

```
  attachInterrupt(1,isrR,RISING);}
```

```
void speedL(){
```

```
  detachInterrupt(1);      //detaches the interrupt
```

```
  timeL = millis()-oldtimeL;    //finds the time
```

```
  rpmL =(revL/timeL)*60000;    //calculates rpm
```

```

oldtimeL=millis();          //saves the current time
revL=0;
attachInterrupt(1,isrL,RISING);}

void speedR(){
detachInterrupt(0);        //detaches the interrupt
timeR = millis()-oldtimeR;    //finds the time
rpmR=(revR/timeR)*60000;     //calculates rpm
oldtimeR=millis();         //saves the current time
revR=0;
attachInterrupt(0,isrR,RISING);}

void loop() {
  if(v<60000){v++;d=d+rpmL; den=den+rpmR;} else
  {rpmL=d*0.0003;rpmR=den*0.0003;d=0; den=0;v=0;

  potValue_1 = analogRead(A0); // reads the value of the
  potentiometer (value between 0 and 1023)

  potValue_2 = ((1023 - potValue_1));

  multiplier_1 = map(potValue_2, 0, 1023, 0, 1000); // scale it to
  use it with the servo library (value between 0 and 180)

  multiplier_2 = map(potValue_1, 0, 1023, 0, 1000); // scale it to
  use it with the servo library (value between 0 and 180)

  motor_speed = analogRead(A1);

```

```
motor_speed = map(motor_speed, 0, 1023, 0, 20000);

speedL();
speedR();
rpmR=rpmR*0.95;

// lcd.init();
// lcd.backlight();
// lcd.setCursor(0,0);
// lcd.print("RPML = ");
// lcd.print(rpmL);
// lcd.setCursor(0,1);
// lcd.print("RPMR = ");
// lcd.print(rpmR);

k=motor_speed*multiplier_1/1000-rpmL;
z1=fuzzy1();
ESC1.write(z1);

l=motor_speed*multiplier_2/1000-rpmR;
z2=fuzzy2();
ESC2.write(z2);

Serial.print("\t reference = ");
```

```
Serial.print(motor_speed);
Serial.print("\t L = ");
Serial.print((float)(motor_speed*multiplier_1/1000));
Serial.print("\t R = ");
Serial.print((float)(motor_speed*multiplier_2/1000));
Serial.print(" RPM_L= ");
Serial.print(rpmL);
Serial.print(" RPM_R= ");
Serial.print(rpmR);
Serial.print(" Error_L= ");
Serial.print(k);
Serial.print(" ErrorR= ");
Serial.print(l);
Serial.print(" MV_L= ");
Serial.print(z1);
Serial.print(" MV_R= ");
Serial.println(z2);}
}
```

--- halaman ini sengaja dikosongkan ---

Lampiran Alat



--- halaman ini sengaja dikosongkan ---

Biodata Penulis



Penulis, lahir di Kota Sukoharjo pada tanggal 22 November 1996, merupakan anak ketiga dari tiga bersaudara. Penulis menyelesaikan pendidikan dasar di SD Ponowaren 03 Tawangarsari, lalu melanjutkan pendidikan tingkat menengah pertama di SMPN 1 Tawangarsari dan Pendidikan tingkat menengah atas di SMAN 1 Sukoharjo dengan jurusan ilmu pengetahuan alam. Penulis memulai kehidupan perkuliahan pada tahun 2015 di S1 Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember di Surabaya. Selama masa perkuliahan penulis aktif di bidang teknologi dan juga mendalami seni budaya dengan bergabung dengan unit kegiatan mahasiswa (UKM) Tari dan karawitan dengan divisi karawitan. Selain itu, penulis juga menjadi asisten praktikum di bidang studi teknik sistem tenaga di laboratorium konversi energi.

Email : sigitarmiyanto11@gmail.com

HP/WA : 082143487159

Line : sigitarmy19