



TUGAS AKHIR - TF141581

**RANCANG BANGUN SISTEM Pengereman
OTOMATIS BERBASIS FUZZY PID PADA
PROTOTYPE MOBIL OTONOM ACePITS**

**DENI AGPRIANTA
NRP. 0231144000006**

**Dosen Pembimbing
Andi Rahmadiansah S.T, M.T.**

**PROGRAM STUDI S1 TEKNIK FISIKA
DEPARTEMEN TEKNIK FISIKA
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2018**

Halaman ini sengaja dikosongkan.



FINAL PROJECT - TF141581

**DESIGN OF AUTOMATIC BRAKING SYSTEM
BASED ON FUZZY-PID AT PROTOTYPE
ACePITS AUTONOMOUS CAR**

**DENI AGPRIANTA
NRP. 0231144000006**

Supervisor
Andi Rahmadiansah S.T, M.T.

**STUDY PROGRAM S1 ENGINEERING PHYSICS
DEPARTMENT OF ENGINEERING PHYSICS
Faculty of Industrial Technology
Sepuluh Nopember Institute of Technology
Surabaya 2018**

Halaman ini sengaja dikosongkan.

PERNYATAAN BEBAS PLAGIASI

Saya yang bertanda tangan di bawah ini:

Nama : Deni Agprianta
NRP : 02311440000006
Departemen/Prodi : Teknik Fisika/ S-1 Teknik Fisika
Fakultas : Fakultas Teknologi Industri
Perguruan Tinggi : Institut Teknologi Sepuluh
Nopember

Dengan ini menyatakan bahwa Tugas Akhir saya yang berjudul “RANCANG BANGUN SISTEM Pengereman Otomatis Berbasis Fuzzy-PID pada Prototipe Mobil Otonom ACePITS” adalah benar karya saya sendiri dan bukan plagiat dari karya orang lain. Apabila dikemudian hari terbukti terdapat plagiat pada Tugas Akhir ini maka saya bersedia menerima sanksi sesuai dengan ketentuan yang berlaku.

Demikian pernyataan ini saya buat dengan penuh tanggung jawab.

Surabaya, 6 Juni 2018
Yang membuat
pernyataan,

Deni Agprianta
NRP. 02311440000006

Halaman ini sengaja dikosongkan.

LEMBAR PENGESAHAN

RANCANG BANGUN SISTEM Pengereman OTOMATIS BERBASIS FUZZY-PID PADA PROTOTYPE MOBIL OTONOM ACePITS

TUGAS AKHIR

Oleh :

Deni Aprianta

NRP : 0231144000006

Surabaya, 6 Juni 2018

Mengetahui

Dosen Pembimbing



Andi Rahmadiansah S.T., M.T.

NIPN. 19790517 200312 1 002

Menyetujui,

Kepala Departemen Teknik Fisika FTI-ITS



Agus Muhammad Hatta, ST, Msi, Ph.D

NIPN. 19780902 200312 1 002

Halaman ini sengaja dikosongkan.

LEMBAR PENGESAHAN

RANCANG BANGUN SISTEM Pengereman OTOMATIS BERBASIS FUZZY-PID PADA PROTOTIPE MOBIL OTONOM ACePITS

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Teknik
pada
Bidang Studi Rekayasa Instrumentasi dan Kontrol
Program Studi S-1 Departemen Teknik Fisika
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember

Oleh :

DENI AGPRIANTA
NRP. 0231144000006

Disetujui oleh Tim Penguji Tugas Akhir :

1. Andi Rahmadiansah S.T., M.T. (Pembimbing)
2. Dr. Suyanto, M.T. (Ketua Penguji)
3. Prof. Dr. Ir Aulia Siti Aisjah, M.T (Penguji)

SURABAYA
JUNI, 2018

Halaman ini sengaja dikosongkan.

**RANCANG BANGUN SISTEM PENEREMAN OTOMATIS
BERBASIS FUZZY-PID
PADA PROTOTYPE MOBIL OTONOM ACePITS**

Nama Mahasiswa : Deni Agprianta
NRP : 0231144000006
Departemen : Teknik Fisika FTI-ITS
Dosen Pembimbing : Andi Rahmadiansah S.T, M.T.

Abstrak

Indonesia merupakan negara dengan mobilitas serta kebutuhan akan transportasi massal yang sangat tinggi. Dalam dekade ini mengembangkan teknologi mobil autonomous berbasis kecerdasan buatan semakin berkembang pesat. Untuk menunjang mobil berbasis kecerdasan buatan diperlukan berbagai komponen pendukung. Salah satu komponen penting itu adalah sistem pengereman. Dalam tugas akhir ini dilakukan rancang bangun sistem pengereman otomatis pada prototype mobil ACePITS. Sistem pengereman otomatis ini dirancang menggunakan 2 buah Motor BLDC 1 KW sebagai penggerak mobil, 1 Kelly KBL kontroler motor, pengereman dilakukan dengan rem cakram hidrolik, yang digerakan oleh servo motor. Untuk perintah pengeremannya berdasarkan kendali rpm dari sensor hall effect sebagai parameter kendali. Kontroler fuzzy-PID dari sistem ini ditanamkan pada Arduino. Dalam tugas akhir ini dilakukan uji sistem pengereman. Berdasarkan data implementasi Fuzzy-PID sistem pengereman otomatis dapat ditanamkan pada kontroler pengereman. Hasil pengujian menggunakan parameter K_p , K_i dan K_d masing masing sebesar 50, 22.5 dan 0.392 didapatkan respon yang relatif stabil serta dapat menghentikan gerak mobil pada saat uji pengereman. Uji pengereman dilakukan dengan variasi kecepatan awal 180 rpm, 200 rpm dan 220 rpm, hasil pengujian didapatkan jarak berhenti prototipe mobil ini masing masing 2,2 meter , 2,8 meter dan 3,4 meter.

Kata kunci: Otomatis, Fuzzy, Pengereman

Halaman ini sengaja dikosongkan

**DESIGN OF AUTOMATIC BRAKING SYSTEM
BASED ON FUZZY-PID
AT PROTOTYPE OF ACePITS AUTONOMOUS CAR**

Nama Mahasiswa : Deni Agprianta
NRP : 0231144000006
Departemen : Teknik Fisika FTI-ITS
Dosen Pembimbing : Andi Rahmadiansah S.T, M.T.

Abstract

Indonesia is a country with mobility and the need for mass transportation is very high. In this decade the development of autonomous car technology based on artificial intelligence is growing rapidly. To support artificial intelligence-based car required various supporting components. One of the important components is the braking system. In this final project is done design of automatic braking system on ACePITS car prototype. Automatic braking system is designed using 2 pieces BLDC 1 KW Motor as the driving force of the car, 1 Kelly KBL motor controller, braking is done with hydraulic disc brakes, driven by servo motor. For braking commands based on the rpm control of the hall effect sensor as a control parameter. The fuzzy- PID of the system is implanted in the Arduino. In this final project conducted braking system test that has been made. Based on Fuzzy-PID implementation data, automatic braking system can be embedded on the controller of the braking. Test results using K_p , K_i and K_d parameters respectively of 50, 22.5 and 0.392 obtained a relatively stable response and can stop the motion of the car during the braking test. Braking test is carried out with an initial speed variation of 180 rpm, 200 rpm and 220 rpm, the test results obtained the distance to stop the prototype of each car is 2.2 meters, 2.8 meters and 3.4 meters.

Keyword: automatic, fuzzy, braking, system

This page is left blank

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT, karena rahmat dan hikmat-Nya sehingga penulis diberikan kesehatan, kemudahan, dan kelancaran dalam menyusun laporan Tugas Akhir yang berjudul:

**“DESIGN OF AUTOMATIC BRAKING SYSTEM
BASED ON FUZZY-PID
AT PROTOTYPE ACePITS AUTONOMOUS CAR”**

Tugas akhir ini merupakan salah satu persyaratan akademik yang harus dipenuhi dalam Program Studi S-1 Teknik Fisika FTI-ITS. Penulis menyampaikan terima kasih yang sebesar-besarnya kepada:

1. Agus M. Hatta, S.T., M.Si., Ph.D. selaku ketua Departemen Teknik Fisika - ITS.
2. Andi Rahmadiansah, ST.,MT. selaku dosen pembimbing tugas akhir ini, yang selalu memberikan bimbingan dan semangat pada penulis.
3. Segenap Bapak/Ibu dosen pengajar di Departemen Teknik Fisika - ITS.
4. Kedua orang tua serta keluarga, yang selalu memberikan dukungan moral, material serta spiritual.
5. Teman-teman asisten Laboratorium Simulasi dan Komputasi yang telah memberikan dukungan dan doa untuk penulis
6. Teman-teman F49 yang telah memberikan dukungan kepada penulis

Penulis menyadari bahwa mungkin masih ada kekurangan dalam laporan ini, sehingga kritik dan saran penulis terima. Semoga laporan ini dapat berguna dan bermanfaat bagi penulis dan pihak yang membacanya.

Surabaya, 6 Juni 2018

Penulis

Halaman ini sengaja dikosongkan.

DAFTAR ISI

HALAMAN JUDUL	i
<i>TITLE PAGE</i>	iii
PERNYATAAN BEBAS PLAGIASI.....	v
LEMBAR PENGESAHAN I.....	vii
LEMBAR PENGESAHAN II.....	ix
Abstrak	xi
Abstract.....	xiii
KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR.....	xix
DAFTAR TABEL	xxi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan.....	2
1.4 Batasan Masalah	3
1.5 Sistematika Laporan	3
BAB II TINJAUAN PUSTAKA.....	5
2.1 Autonomous Car.....	5
2.2 Sistem Pengereman.....	11
2.3 PID (<i>Proportional Integral Derivative</i>).....	14
2.4 Sistem <i>Fuzzy</i>	19
BAB III METODE PENELITIAN.....	23
3.1 Alur Penelitian.....	23
3.2 Studi Litratur	25
3.3 Perancangan <i>Hardware</i>	26
3.4 Perancangan Kontroler	33
3.5 Perancangan Logika Fuzzy pada Matlab	34
BAB IV ANALISA DATA DAN PEMBAHASAN	43
4.1 Pengujian Logika Fuzzy	43
4.2 Hasil Pengujian Sistem	47
BAB V PENUTUP	53
5.1 Kesimpulan.....	53

5.2 Saran.....	53
Daftar Pustaka.....	55
LAMPIRAN A	59
BIOGRAFI PENULIS	75

DAFTAR GAMBAR

Gambar 2.1 Desain Mobil Autonomous.....	5
Gambar 2.2 Kontroler Arduino Mega 2560.	8
Gambar 2.3 <i>Servo Motor</i>	9
Gambar 2.4 <i>Brushless- DC Motor</i>	11
Gambar 2.5 Diagram Blok Sistem Pengendalian	15
Gambar 2.6 Diagram Blok Sistem Kontrol PID.....	17
Gambar 2.7 Arsitektur Sistem <i>Fuzzy</i>	20
Gambar 3.1 Diagram alir penelitian.	24
Gambar 3. 2 Desain 3D rangka mobil ACEPITS.....	26
Gambar 3. 3 HUB motor BLDC 1KW.....	27
Gambar 3. 4 Panel depan Kelly kontroler BLDC.....	28
Gambar 3. 5 Arduino Mega 2560	29
Gambar 3. 6 Servo Towerpro 5010.....	30
Gambar 3. 7 Penempatan Komponen.....	31
Gambar 3. 8 Wiring Sistem Pengereman	32
Gambar 3. 9 Diagram Block Sistem Pengendalian	33
Gambar 3. 10 <i>Interface software</i> Arduino IDE	34
Gambar 3. 11 Diagram Blok Logika Fuzzy	35
Gambar 3.12 FIS untuk Pengereman	36
Gambar 3. 13 Fungsi Keanggotaan <i>Input Error</i>	37
Gambar 3. 14 Fungsi Keanggotaan <i>Input Delta Error</i>	38
Gambar 3.15 Fungsi Keanggotaan <i>Output Kp</i>	39
Gambar 3.16 Fungsi Keanggotaan <i>Output Ki</i>	40
Gambar 3. 15 Fungsi Keanggotaan <i>Output Kd</i>	41
Gambar 4.1 $Input\ error = 0\ \delta\ error = 0$	43
Gambar 4.2 $Input\ error = 0\ \delta\ error = 18$	44
Gambar 4.3 $Input\ error = 0\ \delta\ error = -18$	45
Gambar 4.4 $Input\ error = 12\ \delta\ error = 0$	46
Gambar 4. 5 $Input\ error -12\ dan\ \delta\ error\ 0$	47
Gambar 4.6 Hasil Uji Tanpa dilakukan Pengereman	48
Gambar 4.7 Hasil Uji Tanpa dilakukan Pengereman	49
Gambar 4.8 Hasil Uji Pengereman Kecepatan Awal 200rpm50	
Gambar 4.9 Hasil Uji Pengereman Kecepatan Awal 220 rpm51	

Halaman ini sengaja dikosongkan.

DAFTAR TABEL

Tabel 2. 1 Level Kemudi Otomatis	7
Tabel 2. 2 Spesifikasi Arduino Mega 2560).....	8
Tabel 2. 3 Pengaruh perubahan parameter PID.....	16
Tabel 2. 4 Pengaruh Parameter Kendali PID.....	18

Halaman ini sengaja dikosongkan.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Indonesia merupakan negara kepulauan yang sangat besar. Indonesia terdiri dari 17.499 pulau, terbentang dari Sabang hingga Merauke. Luas total wilayah Indonesia adalah 7,81 juta km² yang terdiri dari 2,01 juta km² daratan, 3,25 juta km² lautan, dan 2,55 juta km² Zona Ekonomi Eksklusif (ZEE) (WARTA-BPHN, 2015). Luasnya wilayah daratan yang besar, Masyarakat Indonesia memerlukan sarana transportasi darat yang memadai. Berdasarkan data yang diambil dari OICA (Organisation Internationale des Constructeurs d'Automobiles), pada tahun 2010, Indonesia tercatat sebagai negara dengan peringkat 21 dalam produksi kendaraan bermotor, dengan jumlah produksi sebesar 702.508 unit dalam satu tahun (OICA, 2011) .

Mobil merupakan salah satu sarana transportasi yang menunjang perekonomian masyarakat Indonesia. Kebutuhan akan mobil semakin meningkat setiap tahunnya. Pada tahun 2016, total penjualan mobil di Indonesia mencapai 1.061.000 unit atau meningkat 4,5% dari tahun 2015 (TRIBUNNEWS, 2017). Perkembangan teknologi membuat produsen mobil menyesuaikan diri untuk membuat mobil-mobil canggih masa depan. Pengembangan teknologi kendaraan yang berbasis kecerdasan buatan ataupun mobil autonomous telah dilakukan. Mobil autonomous merupakan sebuah kendaraan yang dapat bergerak sendiri tanpa adanya pengemudi mobil. Berbagai perusahaan besar dinegara maju tertarik untuk melakukan riset mengenai mobil autonomous. Salah satunya adalah Waymo (anak perusahaan Google) yang pada 7 November 2017 mengumumkan telah melakukan pengujian mobil tanpa sopir diposisi kemudi (Hawkins, 2017). Dengan perkembangan teknologi mobil yang semakin canggih, maka perlu ditingkatkan sistem keamanannya. Salah satu bagian yang vital

untuk meningkatkan keamanan pada mobil adalah sistem pengereman yang baik. Perkembangan teknologi rem mobil berbasis kecerdasan buatan seperti Sistem Fuzzy, PID, Jaringan Syaraf Tiruan dan metode optimisasi semakin berkembang pesat. Pada tahun 2009, telah dilakukan simulasi mengenai sistem pengereman hidrolik menggunakan Fuzzy-PID dan PID konvensional, dalam penelitian tersebut didapatkan bahwa pengereman menggunakan Fuzzy-PID memiliki respon sinyal yang lebih baik dibandingkan dengan penggunaan pengendalian PID konvensional (Junxia Li, 2009). Kemudian penelitian pada tahun 2014, tentang perancangan sistem kendali pengereman otomatis menggunakan jaringan syaraf tiruan untuk melakukan pengereman darurat yang dilakukan oleh Eneh I.I dan Okafor P.U (Eneh II, 2014). Kedua penelitian tersebut masih dilakukan dalam simulasi. Oleh karena itu dalam tugas akhir ini dilakukan sebuah rancang bangun Sistem Pengereman Otomatis pada prototype mobil ACePITS menggunakan Fuzzy-PID.

1.2 Rumusan Masalah

Adapun permasalahan yang muncul dari latar belakang tersebut adalah:

- Bagaimana menentukan spesifikasi komponen yang sesuai untuk sistem pengereman otomatis pada prototipe mobil otonom ACePITS?
- Bagaimana desain sistem pengereman menggunakan metode Fuzzy PID sesuai untuk diterapkan pada mobil otonom ACePITS?

1.3 Tujuan

Tujuan utama dari penelitian ini adalah:

- Menentukan spesifikasi komponen yang sesuai untuk sistem pengereman otomatis pada prototipe mobil otonom ACePITS.

- Mampu mendesain sistem pengereman otomatis menggunakan metode Fuzzy PID untuk diterapkan pada mobil otonom ACePITS.

1.4 Batasan Masalah

Adapun Beberapa batasan masalah pada penelitian ini adalah:

- *Fuzzy Interference System* pada penelitian ini hanya digunakan untuk melakukan *tuning gain* PID pada pengendalian.
- Sistem pengendalian yang dirancang untuk pengereman dengan asumsi mobil bergerak lurus .
- Sistem pengereman otomatis dirancang untuk implementasi mobil ACePITS.
- Variable kendali yang digunakan untuk sistem pengereman adalah rpm.

1.5 Sistematika Laporan

Secara sistematis, penyusunan laporan tugas akhir ini tersusun dalam lima bab dengan penjelasan sebagai berikut:

BAB I Pendahuluan

Bab ini berisi latar belakang, perumusan masalah, tujuan penelitian, batasan masalah, dan sistematika laporan.

BAB II Tinjauan Pustaka

Bab ini berisi mengenai teori-teori penunjang yang terkait dalam penulisan tugas akhir.

BAB III Metodologi Penelitian

Bab ini akan dijelaskan mengenai langkah-langkah yang telah dilakukan dalam penelitian.

BAB IV Analisis Data dan Pembahasan

Bab ini akan ditampilkan data dan analisa hasil simulasi beserta pembahasannya.

BAB V Kesimpulan dan Saran

Bab ini berisi tentang kesimpulan pokok dari seluruh rangkaian penelitian yang telah dilakukan dan saran yang dapat dijadikan sebagai pengembangan penelitian selanjutnya.

Halaman ini sengaja dikosongkan.

BAB II

TINJAUAN PUSTAKA

2.1 Autonomous Car

Autonomous Car merupakan sebuah kendaraan yang dapat bergerak sendiri tanpa adanya seorang pengemudi mobil. Seiring berkembangnya teknologi pada saat ini telah diciptakan berbagai macam sistem otomatis untuk menggantikan peran manusia yang salah satunya pada kendaraan .

Autonomous car dibuat menggunakan berbagai macam teknik untuk mendeteksi objek yang ada disekitarnya, seperti radar, sinar laser, GPS, odometri dan penglihatan komputer. Sistem kontrol yang canggih menginterpretasikan informasi sensor untuk mengidentifikasi jalur navigasi yang sesuai. Autonomous Car ini harus memiliki sistem kontrol yang mampu menganalisa data sensor untuk membedakan antara mobil yang berbeda dengan objek lain yang ada di jalan. Manfaat dari autonomous car meliputi berkurangnya mobilitas dan infrastruktur, peningkatan keamanan, peningkatan mobilitas, peningkatan pelanggan dan berkurangnya kejahatan. Secara khusus penurunan yang paling signifikan yaitu pada kecelakaan lalu lintas.



Gambar 2.1 Desain Mobil Autonomous.

Berdasarkan standar J3016 SAE International yang dipublikasi pada tahun 2014, klasifikasi sistem otomatis dibedakan ke dalam 6 level (mulai dari *full manual* hingga *full automatic*). Klasifikasi ini dilakukan berdasarkan intervensi dan perhatian dari pengemudi. Definisi ini dibuat untuk persyaratan terkait sistem penggerak otomatis kendaraan bermotor di jalan raya, memberikan sebuah sistem klasifikasi yang harmonis dan definisi pendukung (SAE International, 2014).

Level 0 : Sistem otomatis berupa sebuah peringatan tetapi sistem tidak memiliki kendali kendaraan berkelanjutan.

Level 1 : Pengemudi dan sistem otomatis berbagi kendali atas kendaraan. Contohnya seperti *Adaptive Cruise Control (ACC)*, di mana pengemudi mengendalikan kemudi dan sistem otomatis mengontrol kecepatan dan Parkir Bantuan, di mana kemudi otomatis sementara kecepatan berada di bawah kendali manual. Pengemudi harus siap untuk mengambil alih kendali penuh kapan saja.

Level 2 : Sistem otomatis mengambil kendali penuh atas kendaraan (akselerasi, pengereman, dan kemudi). Sopir harus memantau mengemudi dan siap untuk mengambil alih kendali segera setiap saat jika sistem otomatis gagal merespons dengan benar.

Level 3 : Pada sistem otomatis ini, pengemudi dapat dengan aman mengalihkan perhatian mereka dari tugas mengemudi, misalnya pengemudi dapat mengirim teks atau menonton film. Kendaraan akan menangani situasi yang memerlukan tanggapan cepat, seperti pengereman darurat. Pada sistem ini pengemudi tetap harus siap mengambil alih kemudi dalam waktu tertentu.

Level 4 : Seperti level 3, sistem ini tidak memerlukan perhatian pengemudi, pengemudi dapat dengan aman pergi tidur atau meninggalkan kursi pengemudi. Sistem dapat berkendara sendiri hanya pada area spasial terbatas (geofenced) atau dalam keadaan khusus, seperti kemacetan lalu lintas.

Tabel 2. 1 Level Kemudi Otomatis (SAE International, 2014)

Level	Nama	Steering, Acceleration, Deceleration	Monitoring of Driving Environment	Fallback Performance of dinamic Driving Task
0	No Automation	Human	Human	Human
1	Driver Assistence	Human and System	Human	Human
2	Partial Automation	System	Human	Human
3	Conditional Automation	System	System	Human
4	High Automation	System	System	System
5	Full Automation	System	System	System

Di luar area atau keadaan ini, kendaraan harus dapat membatalkan perjalanan dengan aman, yaitu memarkir mobil, jika pengemudi tidak mengambil alih kemudi.

Level 5 : Pada sistem ini tidak ada intervensi manusia yang diperlukan sama sekali.

Tabel 2.1 menunjukkan klasifikasi level sistem otomatis pada Kendaraan berdasarkan pembagian peran dari pengemudi dan sistem.

2.1.1 Mikrokontroler

Mikrokontroler adalah salah satu bagian dasar dari sistem komputer. Meskipun mempunyai bentuk yang jauh lebih kecil dari suatu komputer pribadi dan komputer mainframe, mikrokontroler dibangun dari elemen-elemen dasar yang sama. Artinya, bagian terpenting dan utama dari suatu sistem terkomputerisasi adalah program itu sendiri yang dibuat oleh seorang programmer.

Tabel 2. 2 Spesifikasi Arduino Mega 2560 (Arduino, 2018)

Spesifikasi	Keterangan
<i>Flash Memory</i>	256 KB
Tegangan Operasi	5V
Digital I/O Pins	54 (15 pinnya mendukung PWM output)
Arus DC per I/O pin	20mA
<i>Clock Speed</i>	16 MHz
SRAM/EEPROM	8KB/4KB
Dimensi	101.52 x 53.3 mm ²

Program ini menginstruksikan komputer untuk melakukan jalinan yang panjang dari aksi-aksi sederhana untuk melakukan tugas yang lebih kompleks yang diinginkan oleh programmer. Sistem mikrokontroler tersebut dapat digunakan untuk akuisisi data seperti mikrokontroler Teensy.

Arduino Mega merupakan rangkaian elektronik yang bersifat *open-source*, serta memiliki perangkat keras dan lunak yang mudah untuk digunakan, berbasis ATmega2560. Arduino Mega mempunyai 54 pin *input/output* digital dengan 15 pin dapat digunakan untuk keluaran PWM, 16 input analog, 4 UART (*hardware serial port*) (Arduino, 2018). Tabel 2.2 merupakan tabel spesifikasi Mikrokontroler Arduino Mega 2560.



Gambar 2.2 Kontroler Arduino Mega 2560. (Arduino, 2018)

Salah satu fitur yang ada pada Arduino adalah dilengkapi dengan 15 pin yang dapat memberikan output PWM. *Pulse Width Modulation* merupakan dasar untuk control dalam elektronika. Secara teori, *zero rise* dan *fall time* dari sebuah gelombang PWM ideal untuk mengendalikan daya dari perangkat semikonduktor. Sebagian besar sirkuit elektronik dikendalikan oleh sinyal PWM dari berbagai bentuk. Sinyal naik dan turun yang cepat dapat memastikan bahwa perangkat semikonduktor dihidupkan atau dimatikan secepat mungkin untuk meminimalkan waktu transisi *switching*. Frekuensi pulsa merupakan parameter yang paling penting saat mendefinisikan metode PWM.

2.1.2 Servo

Aktuator merupakan alat yang berguna untuk mengontrol tarikan mekanik dari system pengereman. Motor servo adalah sebuah perangkat atau aktuator (motor) yang dirancang dengan sistem kontrol *feed back loop* tertutup (servo), sehingga dapat di atur untuk menentukan posisi sudut dari poros *output* motor. Motor servo terdiri dari motor DC, serangkaian gear, rangkaian kontrol dan potensiometer.



Gambar 2.3 Servo Motor (Adafruit, n.d.).

2.1.3 BLDC Motor

Sistem Penggerak mobil listrik pada umumnya menggunakan motor BLDC. *Brushless* DC Motor termasuk kedalam jenis motor sinkron. Artinya medan magnet yang dihasilkan oleh stator dan medan magnet yang dihasilkan oleh rotor berputar pada frekuensi yang sama. Motor BLDC tidak mengalami slip seperti yang terjadi pada motor induksi biasa. Motor jenis ini mempunyai magnet permanen pada bagian rotor dan elektromagnet pada bagian stator. Setelah itu, dengan menggunakan sebuah rangkaian sederhana (*simple computer system*), maka kita dapat merubah arus di elektromagnet ketika bagian rotornya berputar.

BLDC motor merupakan motor listrik sinkron AC 3 fasa, motor ini disebut dengan BLDC karena pada implementasinya BLDC menggunakan sumber DC sebagai sumber energi utama yang kemudian diubah menjadi tegangan AC dengan menggunakan inverter 3 fasa.

Tujuan dari pemberian tegangan AC 3 fasa pada stator BLDC adalah menciptakan medan magnet putar stator untuk menarik magnet rotor. Oleh karena tidak adanya *brush* pada motor BLDC, untuk menentukan *timing* komutasi yang tepat pada motor ini sehingga didapatkan torsi dan kecepatan yang konstan, diperlukan 3 buah sensor hall.

Keunggulan dari *brushless* DC motor adalah sebagai berikut:

- Kecepatan putaran *motor* dapat diatur oleh mikrokontroler sehingga dapat menjaga efisiensi dari *brushless motor*.
- Tidak adanya *storing/electrical noise*.
- Tidak menggunakan *brushes* yang dapat rusak seiring bertambahnya usia pemakaian *motor*.
- Dengan posisi *electromagnets* di bagian *stator*, maka pendinginan motor menjadi lebih mudah.



Gambar 2.4 *Brushless- DC Motor* (kellycontroller, 2018).

2.2 Sistem Pengereman

Sistem rem adalah alat yang digunakan untuk memperlambat atau menghentikan laju kendaraan. Adanya rem pada kendaraan sangat penting untuk keselamatan pengendara, jika tidak ada rem maka pengendara bisa mengalami kecelakaan. Rem merupakan salah satu dari bagian kendaraan yang mempunyai peranan penting untuk kenyamanan dan keselamatan pengendara sepeda motor. Gerak roda yang diperlambat, otomatis gerak kendaraan menjadi lambat. Fungsi rem adalah menyerap energi kinetik dari bagian yang bergerak atau energi potensial yg ditimbulkan oleh komponen lain (K.M.Jossy, 2011). Dengan kata lain rem adalah komponen yang mengubah energi mekanik menjadi energi thermal / panas melalui gesekan. Selain itu rem adalah komponen yang sangat penting bagi kendaraan / alat transportasi. Semakin cepat suatu kendaraan maka memerlukan kapasitas sistem pengereman yang efektif pula. Menurut K.M.Jossy, Kapasitas rem tergantung oleh faktor–faktor berikut :

- Tekanan antara kampas rem dengan permukaan bidang pengereman.
- Koefisien gesek antara kampas rem dengan bidang pengereman.
- Batasan kecepatan motor.
- Area yang terkena gesekan (bidang gesek).

Kemampuan kampas rem menyerap panas yang ditimbulkan oleh gesekan. Sistem pengereman mobil dirancang untuk mengontrol kecepatan/laju (mengurangi/ memperlambat kecepatan dan menghentikan laju) mobil, dengan tujuan meningkatkan keselamatan dan untuk memperoleh pengendalian yang aman. Prinsip kerja rem adalah dengan mengubah energi gerak/kinetik menjadi energi panas dalam bentuk gesekan. Berikut macam-macam rem menurut konstruksinya, yaitu rem tromol (drum brake), dan rem cakram (disc brake).

2.2.1 Rem tromol

Rem tromol adalah salah satu konstruksi rem yang cara pengereman kendaraan dengan menggunakan tromol rem (brake drum), sepatu rem (brake shoe), dan silinder roda (wheel cylinder). Pada dasarnya jenis rem tromol yang digunakan roda depan dan belakang tidak sama, hal ini dimaksudkan supaya sistem rem dapat berfungsi dengan baik dan sesuai dengan persyaratan (Andun, Adhari, Agus, 2005:12). Pada rem tromol, kekuatan tenaga pengereman diperoleh dari sepatu rem yang diam menekan permukaan tromol yang berputar bersama dengan roda. Rem tromol mempunyai keuntungan dibandingkan dengan tipe rem cakram, yaitu adanya self energizing effect yang memperkuat daya pengereman, hanya saja konstruksinya agak rumit dan tertutup sehingga radiasi

panas ke udara luar dan water recovery kurang baik. Water recovery merupakan kemampuan bidang gesek (sepatu rem/pad) untuk mengembalikan koefisien gesek pada kondisi semula, pada saat sistem rem terkena air yang mengakibatkan koefisien gesek sepatu rem/pad menjadi berkurang karena terlumasi oleh air. Pada saat sistem rem terkena air, tipe rem cakram memiliki kemampuan water recovery yang lebih baik dibandingkan dengan sistem rem tromol, hal ini disebabkan karena air akan terlempar keluar dari permukaan cakram dan pad karena adanya gaya sentrifugal. Pada rem tromol tetap akan menyisakan air diantara sepatu rem dan tromol sehingga koefisien gesek rem menjadi rendah.

2.2.2 Rem Cakram

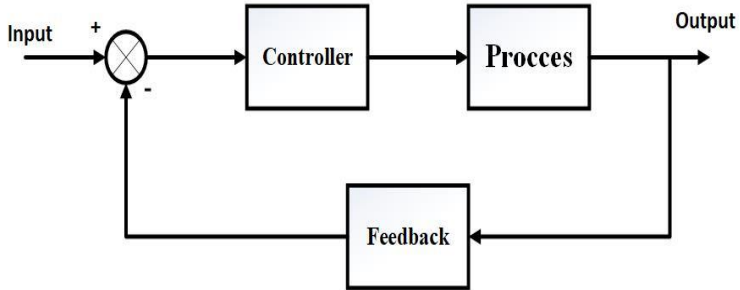
Rem cakram banyak digunakan, hampir di semua produsen otomotif telah menggunakan rem cakram pada sistem pengeremannya dan sudah merupakan standar pada kendaraan model baru. Konstruksi rem cakram pada umumnya terdiri atas cakram (disc rotor) yang terbuat dari besi tuang yang berputar dengan roda, bahan gesek (disc pad) yang menjepit & mencengkeram cakram, serta kaliper rem yang berfungsi untuk menekan & mendorong bahan gesek. Konstruksi pada rem cakram hampir sama dengan rem tromol, dimana tromolnya ditiadakan dan sebagai gantinya dipasang sekeping cakram. Pada rem cakram terdapat sepatu-sepatu rem yang dilengkapi dengan pelapis sepatu rem, cara kerja rem ini secara hidrolis. Daya pengereman terjadi karena adanya gesekan antara cakram dengan pad, sehingga pengereman terjadi (Andun, Adhari, Agus, 2005:12). Salah satu kelemahan pada sistem rem cakram yaitu Self energizing effect yang terjadi pada rem cakram sangat kecil, sehingga diperlukan tekanan pengereman yang lebih besar untuk mendapatkan daya pengereman yang efisien dan pad cenderung lebih cepat aus dibanding dengan sepatu rem pada rem tromol. Disamping kelemahannya tersebut rem cakram mempunyai beberapa kelebihan,

diantaranya : konstruksi sederhana, penggantian pad mudah, tanpa penyetelan, bidang gesek selalu terkena udara sehingga radiasi panasnya sangat baik dan water recovery sangat baik karena air akan terlempar keluar dari permukaan cakram dan pad karena adanya gaya sentrifugal.

Prinsip kerja rem cakram pada dasarnya adalah dengan membuat gaya gesek antara piringan cakram yang terhubung dengan roda kendaraan dengan breake pad yang ditekan oleh piston. secara umum sama dengan rem tromol yaitu menggunakan bahan friksi pada sepatu rem untuk mengurangi atau menghentikan laju kendaraan. Rem ini bekerja dengan dengan menjepit cakram yang biasanya dihubungkan dengan roda kendaraan, serta untuk menjepit cakram digunakan bahan friksi atau kampas rem dalam bentuk sepatu rem dengan mekanismenya diatur oleh kaliper rem. Serta untuk menggerakkan atau mengatur mekanisme caliper rem menggunakan gaya mekanik, hidrolis, pneumatik atau elektronik yang melawan gaya dari kedua sisi cakram.

2.3 PID (*Proportional Integral Derivative*)

Kendali PID telah menjadi pondasi keberhasilan pengendalian beragam variabel proses industri selama lebih dari setengah abad. Kendali PID dapat digunakan hampir pada setiap industri yang bergerak dalam bidang proses. Menurut sebuah survey dinyatakan bahwa 97 % industri yang bergerak dalam bidang proses menggunakan PID dalam pengendaliannya. (Setiawan, 2008). Gambar 2.5 merupakan diagram blok sistem pengendalian.



Gambar 2.5 Diagram Blok Sistem Pengendalian

Kendali PID telah mengalami banyak perubahan dalam teknloginya. Dari mekanik, pneumatic sampai mikroprosesor kendali PID sampai hari ini banyak menggunakan mikroprosesor dalam prakteknya.

Variabel y adalah variabel proses yang diukur, r adalah variabel referensi, u adalah sinyal kendali dan e adalah *error* kendali ($e = y \text{ setpoint} - y$). Variabel referensi juga sering disebut dengan setpoint. Sinyal kendali adalah penjumlahan total dari 3 paramater: P (proportional terhadap *error*), I (Proportional terhadap Integral dari *error*), dan D (proporsional terhadap derivative dari *error*). Sedangkan parameter kendalinya adalah proportional gain K , waktu Integral T_i , dan waktu derivative T_d .

Aspek yang sangat penting dalam desain kendali PID adalah penentuan parameter kendali PID supaya sistem *close loop* memenuhi kriteria performansi yang diinginkan. Hal ini disebut juga dengan *tuning*. Banyak metode yang telah digunakan untuk *tuning* PID diantaranya adalah metode *Ziegler Nichols*, *Root Locus* dan lain sebagainya. Karena nilai dari parameter masing masing K_p , T_i , dan T_d akan berpengaruh pada respon sistem.

Tabel 2. 3 Pengaruh perubahan parameter PID terhadap respon sistem (Masbi, 2015).

Koefisien	Peran	Inisialisasi Besar	Inisialisasi Kecil
Proportional (Kp)	Percepatan, stabilitas	Penurunan stabilitas	Kesalahan permanent, memperlambat respon
Integrator (Ki)	Penurunan kesalahan permanen	Penurunan stabilitas, Perpanjangan respon osilasi	Menuju respon permanen lebih lambat
Derivatif (Kd)	Menaikkan stabilitas, Penurunan jarak osilasi	Menaikkan stabilitas, amplifikasi turbulensi	Kegagalan mendapatkan keuntungan

Pada tabel 2.3 menunjukkan beberapa efek pada masing masing pengendali proportional, integral, dan derivative terhadap respon sistem. Tabel tersebut berguna untuk menentukan koefisien PID

Algoritma dari PID dapat dideskripsikan pada persamaan berikut:

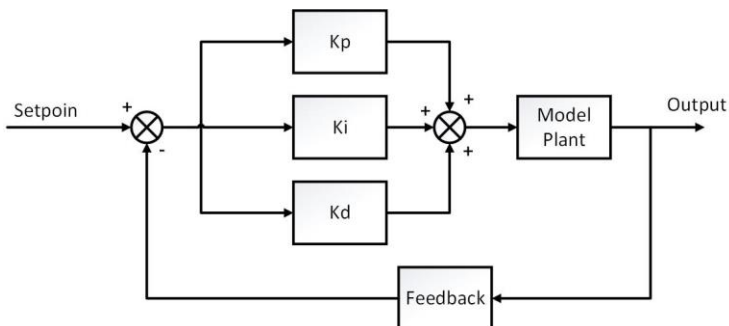
$$u(t) = K \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right) \quad (2.1)$$

dengan y adalah variabel proses terukur, r adalah variabel referensi, u adalah sinyal kendali dan e adalah *error* kendali ($e = y_{sp} - y$). Variabel referensi juga sering disebut dengan *set point*. Sinyal kendali adalah penjumlahan total dari tiga term: *P-term* (proporsional terhadap *error*) *I-term* (proporsional terhadap integral dari *error*), dan *D-term*

(proporsional terhadap derivatif dari *error*). Parameter pengendalinya adalah Proportional gain K_p , waktu Integral T_i , dan waktu *Derivative* T_d .

PID terdiri atas tiga parameter yaitu parameter proporsional (K_p), integral (T_i), dan derivatif (T_d). Komponen proporsional (P) berperan untuk mengalikan sinyal *error* dengan nilai gain secara proporsional. Nilai penguat proporsional yang tinggi akan mengakibatkan perubahan sinyal output yang besar pula, namun nilai gain proporsional yang terlalu besar akan mengakibatkan sistem menjadi tidak stabil. Nilai penguat proporsional yang terlalu kecil maka sinyal keluaran akan menjadi terlalu kecil untuk memberi respon terhadap gangguan.

Komponen integral (I) pada pengendali PID digunakan untuk mengoreksi jika terjadi perbedaan nilai antara sinyal keluaran dengan sinyal setpoint pada kondisi *steady*. Pengendali integral melakukan akumulasi terhadap nilai *error* dari waktu ke waktu di masa lampau dan mengatasi kekurangan dari pengendali proporsional dengan mengeliminasi nilai *error steady-state*.



Gambar 2.6 Diagram Blok Sistem Kontrol PID (Sufendi, Trilaksono, Nasution, & Purwanto, 2013)

Pengendalian derivatif (D) menggunakan tingkat perubahan sinyal *error* sebagai elemen prediksi pada aksi pengendalian. Komponen derivatif tidak dapat digunakan sebagai pengendali yang berdiri sendiri tanpa komponen pengendalian lain akibat adanya kemungkinan peningkatan nilai *noise* dalam pengukuran. Pengendali derivatif akan memberikan pengaruh berupa peningkatan stabilitas sistem, pengurangan nilai overshoot, serta memperbaiki respon transien. Pengaruh komponen pengendalian K_p , K_i , dan K_d pada sistem loop tertutup dirangkum pada Tabel 2.4.

Parameter-parameter pengendalian K_p , T_i , dan T_d biasanya diatur pada batasan nilai tertentu. Kegiatan menentukan nilai parameter K_p , T_i , dan T_d disebut *tuning* parameter pengendalian. Pengaturan nilai parameter harus dilakukan dengan tepat dengan mempertimbangkan sifat dan karakteristik *plant* yang akan dikendalikan untuk menghasilkan respon dan perilaku sistem pengendalian yang diinginkan. (K. A Tehrani, 2012)

Penentuan nilai parameter pengendali merupakan salah satu faktor terbesar yang mempengaruhi kinerja pengendalian PID. Parameter pengendali PID dapat di-*tuning* dengan menggunakan berbagai cara, antara lain *tuning* manual, metode Ziegler-Nichols, metode Cohen-Coons, *loop sharing* serta *pole placement* dan berbagai metode *tuning* lain. (A. A. Khan, 2006).

Tabel 2.4 Pengaruh Parameter Kendali PID (S, 2016)

Parameter	<i>Rise time</i>	<i>Overshoot</i>	<i>Settling time</i>	<i>Steady-state error</i>
K_p	Berkurang	Meningkat	Perubahan kecil	Berkurang
T_i	Berkurang	Meningkat	Meningkat	Berkurang
T_d	Sedikit berkurang	Sedikit berkurang	Sedikit berkurang	Tidak berpengaruh

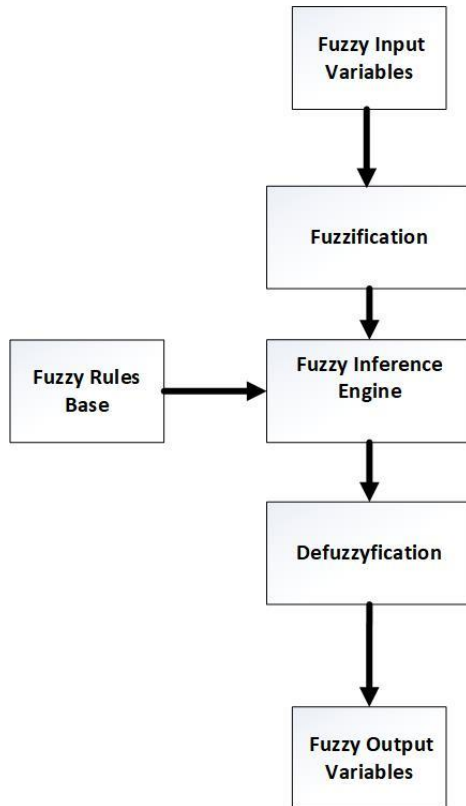
Metode *tuning* yang dikembangkan oleh Ziegler dan Nichols merupakan metode penentuan parameter pengendalian PID yang paling sering digunakan di industri baik dalam bentuk murni maupun sudah dimodifikasi. Metode ini sering digunakan sebagai dasar prosedur *tuning* yang diterapkan oleh manufaktur pengendali dan industri proses. Metode *tuning* Ziegler-Nichols didasarkan atas penentuan beberapa sifat dinamik proses yang kemudian digunakan untuk menyatakan nilai parameter pengendali. Luasnya pengaplikasian metode ini dikarenakan prosedur yang sederhana dan intuitif, memerlukan banyak pengetahuan mengenai proses yang dikendalikan serta dapat diaplikasikan dengan mudah. Beberapa batasan masih dimiliki oleh metode Ziegler-Nichols dalam proses *tuning* parameter pengendali PID, yaitu batas kestabilan yang sempit serta kemampuan untuk mengatasi adanya gangguan eksternal yang kurang memuaskan pada beberapa proses. (K. Astrom, 1995)

2.4 Sistem Fuzzy

Logika *fuzzy* dapat dikonseptualisasikan sebagai generalisasi dari logika klasik untuk menentukan keputusan berbasis sebab dan akibat (*if-then*). Dengan menggunakan logika *fuzzy*, tidak diperlukan model matematis untuk mengetahui karakteristik sistem. Dari pernyataan tersebut, metode ini memiliki beberapa kelebihan diantaranya arsitektur yang lebih sederhana dibandingkan dengan metode lainnya, kekokohan sistem (*robustness*), performansi yang lebih baik, dan lain-lain. Melihat sistem yang bersifat non-linear pada setiap parameter, logika *fuzzy* merupakan metode yang cocok untuk menerapkan algoritma *tuning* PID secara otomatis (Gautam, 2013).

Fuzzy Interference System (FIS) merupakan pemetaan input data vektor secara *non-linear* menjadi sebuah *output*

skalar menggunakan *fuzzy rules*. Proses pemetaan tersebut melibatkan fungsi keanggotaan dari *input/output*, operator logika fuzzy, *fuzzy rules (if-then)*, agregasi dari *set output*, dan defuzzifikasi. FIS terdiri atas empat komponen, yaitu *fuzzifier*, *inference engine*, *rule base*, *defuzzifier*. Rule base terdiri atas aturan linguistik yang berasal dari pakar. Ketika rules sudah ditetapkan, FIS dapat didefinisikan sebagai sistem yang memetakan input vektor menjadi output vektor.



Gambar 2.7 Arsitektur Sistem *Fuzzy*.

2.3.1. Fuzzifier

Dalam logika *fuzzy*, masukan dari keluaran sistem akan dinyatakan dalam fungsi keanggotaan. Fungsi keanggotaan merupakan suatu *range* nilai yang dapat merepresentasikan bentuk masukan atau keluaran sistem. Dalam proses fuzzifikasi ini masukan *fuzzy* akan diubah menjadi variabel linguistik yang akan diolah dalam *inference engine* (Gautam, 2013). Fungsi keanggotaan juga didefinisikan sebagai sebuah kurva untuk memetakan masukan ke derajat keanggotaan *fuzzy* yang memiliki beberapa bentuk fungsi, diantaranya:

- Fungsi Gaussian

$$\mu_A(u) = \exp \left[-\frac{(u-x)^2}{2\sigma^2} \right] \quad (2.2)$$

- Fungsi Segitiga

$$\mu_A(u) = 1 - \frac{\sqrt{(u-a)^2}}{b} \quad (2.3)$$

- Fungsi Trapesium

$$\mu_A(u) = \begin{cases} 1; 0 \geq (u - a) \leq \frac{b}{a} \\ 2 - 2\frac{\sqrt{(u-a)^2}}{b}; \frac{b}{a} \leq (u - a) \leq b \end{cases} \quad (2.4)$$

2.3.2. Inference Engine dan Rule Base

Masukan sistem yang telah melalui proses fuzzifikasi akan diolah dalam sebuah *inference engine*. Di dalam *inference* terdapat aturan-aturan (*rule base*) yang menggunakan metode *fuzzy* Mamdani. *Fuzzy* mamdani merupakan jenis *fuzzy* yang memiliki masukan dan keluaran berupa numerik. Masukan numerik tersebut akan diolah oleh *inference engine* untuk menghasilkan keluaran *fuzzy* dengan komposisi *if-then* yang ada di dalam *rule base*.

2.3.3. Defuzzifier

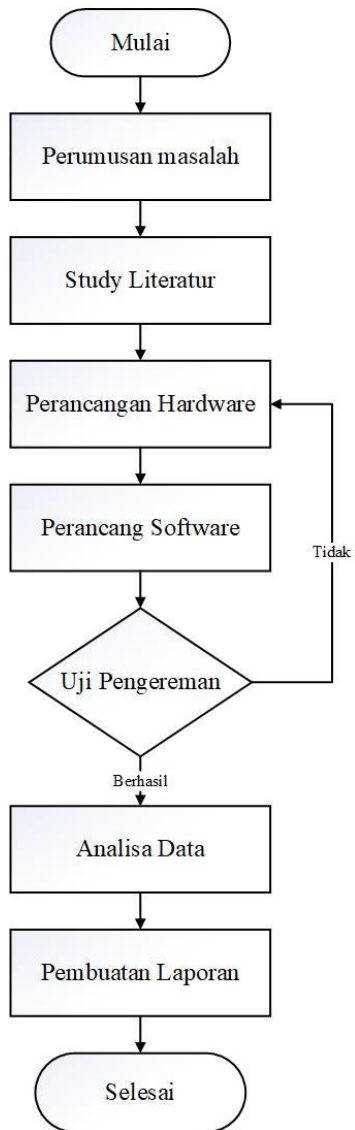
Pada proses defuzzifikasi, variabel linguistik yang merupakan keluaran dari inference engine dirubah menjadi variabel numerik sesuai dengan fungsi keanggotaan keluaran sistem fuzzy tersebut. Angka numerik yang dihasilkan merupakan keluaran dari sistem fuzzy. Ada lima metode dalam defuzzifikasi, yaitu metode max, metode titik tengah, metode rata-rata, metode penjumlahan titik tengah, dan metode titik tengah terbesar.

BAB III METODE PENELITIAN

3.1 Alur Penelitian

Pada penelitian ini dilakukan perancangan mengenai sistem pengereman otomatis menggunakan *Fuzzy-PID*. Di dalam penelitian ini, ada dua *input* yang digunakan pada *Fuzzy*, yaitu *error* dan *delta error*, sementara *output* yang dihasilkan yaitu tiga *parameter* kendali PID, K_p , K_i dan K_d . *Error* merupakan selisih antara *output* yang dihasilkan dengan *setpoint*, sementara *delta error* merupakan selisih dari setiap *error* yang terjadi.

Kedua input *error* dan *delta error* berpengaruh terhadap *gain* PID, dimana setiap *parameter* PID tersebut nantinya akan berpengaruh terhadap performa dari sistem pengereman. Misalnya, apabila *error* yang dihasilkan cukup besar dan *delta error* tidak begitu tinggi, maka *gain* PID akan dituning sedemikian rupa agar sistem dapat stabil kembali dan sistem pengendalian dengan performa yang baik dapat tercapai. Semakin minimal *settling time* dan *overshoot* dari respon, maka performa sistem kendali semakin baik. Agar mendapatkan hasil yang optimal, maka sebelum penelitian ini dimulai, dilakukan terlebih dahulu studi literatur mengenai pengendalian Sistem pengereman dengan *Fuzzy Inference System* sebagai sistem *self tuning gain* PID pada controller yang digunakan. Setelah itu mencari variabel yang akan digunakan dalam sistem pengereman. Lalu merancang hardware *prototype* mobil dan sistem pengeremannya, Setelah itu sistem kontrol *Fuzzy-PID* dirancang untuk diterapkan pada sistem pengereman. Uji kinerja sistem pengereman otomatis pada mobil autonomous diperlukan untuk mengetahui apakah sistem yang dirancang telah berhasil, dapat beroperasi dengan baik dan dapat menghentikan laju mobil. Untuk alur selengkapnya dapat di lihat pada diagram alir yang ditunjukkan Gambar 3.1.



Gambar 3.1 Diagram alir penelitian.

3.2 Studi Literatur

Studi literatur berupa pemahaman secara teoritis mengenai sistem pengereman mulai dari jenis jenis rem pada mobil, model matematis, sampai tahap pemilihan alat untuk implementasi. Selain itu dari segi pengendalian, memahami karakteristik dari sensor, prosesor dan aktuator yang digunakan pada penelitian ini berdasarkan jurnal – jurnal penelitian yang sudah pernah dilakukan dan juga dari datasheet sensor, prosesor, dan aktuator tersebut. Pemahaman tentang pengereman dan parameter apa saja yang berpengaruh terhadap keberhasilan rem tersebut. Kemudian pemahaman tentang strategi kendali untuk pengendalian pengereman, mulai dari membandingkan metode PID, jaringan syaraf tiruan, serta logika fuzzy berdasarkan jurnal jurnal penelitian yang terpercaya. Pada tahun 2009, telah dilakukan penelitian mengenai sistem pengereman hidrolik menggunakan Fuzzy-PID dan PID konvensional, penelitian tersebut dilakukan dengan simulasi menggunakan software Matlab. Hasil dari penelitian tersebut didapatkan bahwa pengereman menggunakan Fuzzy-PID memiliki respon sinyal yang lebih baik dibandingkan dengan penggunaan pengendalian PID konvensional (Junxia Li, 2009). Pada simulasi tersebut sinyal output dari Fuzzy-PID tidak memiliki nilai overshoot dan osilasi yang terjadi lebih rendah dibandingkan dengan metode PID konvensional. Kemudian penelitian pada tahun 2014, tentang perancangan sistem kendali pengereman otomatis menggunakan jaringan syaraf tiruan dengan metode *training back propagation* untuk melakukan pengereman darurat yang dilakukan oleh Eneh I.I dan Okafor P.U (Eneh II, 2014). Pada penelitian tersebut pengereman dapat dilakukan dengan aman. Kedua penelitian tersebut masih dilakukan dalam bentuk simulasi. Fuzzy-PID pada penelitian ini dipilih berdasarkan rekomendasi penelitian sebelumnya. Metode kendali menggunakan Fuzzy ataupun PID dapat diterapkan pada mikrokontroler arduino Pada tahapan tinjauan pustaka,

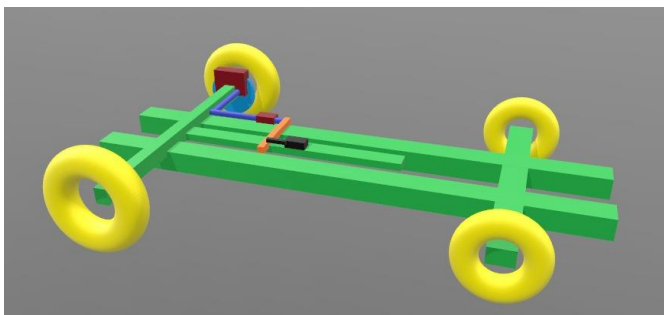
dilakukan kajian dari penelitian sebelumnya yang berkaitan dengan pengeraman dengan metode kendali lain, serta disimpulkan metode kendali yang cocok untuk diimplementasikan pada tugas akhir ini.

3.3 Perancangan *Hardware*

Tahap ini melakukan pembuatan mobil serta perakitan komponen penyusun sistem pengereman otomatis, baik elektrik maupun mekanik. Sistem pengereman dalam penelitian ini menggunakan rem cakram hidrolik yang dihubungkan dengan motor servo sebagai aktuatornya. Pengendalian sistem pengereman otomatis ini dilakukan berdasarkan input variabel kecepatan yang dibaca menggunakan sensor hall effect yang ada pada BLDC motor. Data kecepatan kemudian dibaca oleh mikrokontroler Arduino Mega yang kemudian diproses untuk menggerakkan servo (melakukan pengereman).

3.3.1 Rangka Mobil ACePITS

ACePITS singkatan dari Autonomous Car Engineering Phisyc Institut Teknologi Sepuluh Nopember adalah *prototype* mobil listrik yang didesain berbentuk mobil Gokart dengan ukuran $2 \times 1 \text{ m}^2$. Gambar 3.2 merupakan rancangan desain 3D dari rangka mobil ACePITS. Prototipe mobil ACePITS sendiri memiliki berat kurang lebil 120 kg.



Gambar 3. 2 Desain 3D rangka mobil ACePITS

3.3.2 Motor

Mobil ACePITS menggunakan 2 HUB motor BLDC dengan daya 1000W sebagai penggerak mobil. Hub motor BLDC dengan inputan 48V. Motor BLDC ini memiliki kecepatan maksimal mencapai 45 Km/Jam. HUB motor BLDC dengan ukuran diameter 10 inchi digunakan sebagai roda bagian belakang mobil ACePITS. Pada penelitian ini motor BLDC dipilih karena kecepatan motor ini dapat diatur menggunakan kontroler serta memiliki daya tahan yang lebih baik jika dibandingkan dengan motor brushed. Dengan beban prototipe mobil sebesar 120kg, Gambar 3.3 merupakan bentuk motor BLDC yang digunakan dalam mobil ACePITS.



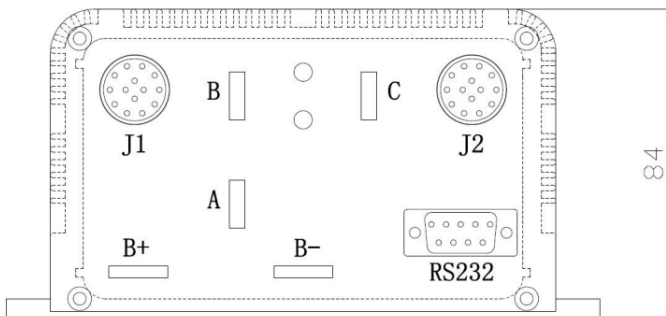
Gambar 3. 3 HUB motor BLDC 1KW

3.3.3 Kontroler BLDC

Kontroler motor BLDC yang digunakan dalam pembuatan mobil ACEPITS menggunakan Kelly KBL48501X untuk mengendalikan putaran dari motor BLDC dengan input tegangan 48V dan arus maksimum sebesar 500 A.

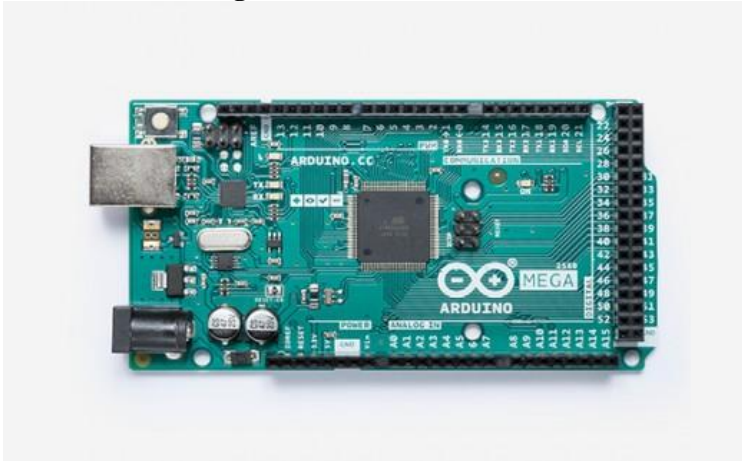
Kontroler Kelly KBL48501X memiliki berbagai pin yang digunakan untuk mengatur input dan outputnya. Pin B+ pada kontroler ini digunakan sebagai masukan baterai positif. Pin B- digunakan sebagai jalur masukan baterai negatif. Pin A digunakan sebagai keluaran fase A dari kontroler. Pin B digunakan sebagai keluaran fase B dari kontroler. Pin C digunakan sebagai keluaran fase C dari kontroler. Pin J1 digunakan untuk melakukan komunikasi dengan kontroler lain dan J2 digunakan untuk mengatur input dan output dari kontroler. Konfigurasi pin yang ada pada Kelly kontroler BLDC di tunjukan pada Gambar 3.4

Pemilihan kontroler kelly BLDC didasarkan pada motor BLDC yang digunakan dalam prototipe mobil listrik. Jika daya motor yang digunakan dalam motor memiliki spesifikasi daya 1 KW dan tegangan input 48 V, maka kontroler yang digunakan harus memiliki tegangan yang sama dan daya yang sama ataupun bisa lebih. Dalam penelitian ini, prototipe mobil ACEPITS menggunakan Kelly KBL48501X sebagai kontroler BLDC.



Gambar 3. 4 Panel depan Kelly kontroler BLDC

3.3.4 Arduino Mega 2568



Gambar 3. 5 Arduino Mega 2560 (Arduino, 2018)

Arduino Mega dalam penelitian ini merupakan kontroler utama yang digunakan untuk mengendalikan pengereman dan memberikan sinyal *input* untuk *trotle* pada kontroler Kelly. Arduino mega dipilih karena merupakan kontroler programabel yang memiliki fitur lengkap, didukung dengan *Input Output* digital ataupun analog, mendukung fitur PWM *output* yang diperlukan dalam mengendalikan perangkat sistem pengereman. Gambar 3.5 menunjukan bentuk fisik dari kontroler arduino mega 2560.

3.3.5 Motor Servo

Aktuator yang digunakan dalam sistem pengereman ini adalah motor servo. Beban tuas rem cakram pada sistem pengereman sebesar 14 kg. Motor servo yang digunakan dalam sistem pengereman ini adalah Tower Pro 5010. Motor servo ini memiliki torsi sebesar 5.5 kg/cm pada tegangan 5V (Adafruit, n.d.). Pesawat sederhana digunakan agar motor servo tersebut dapat menarik tuas.

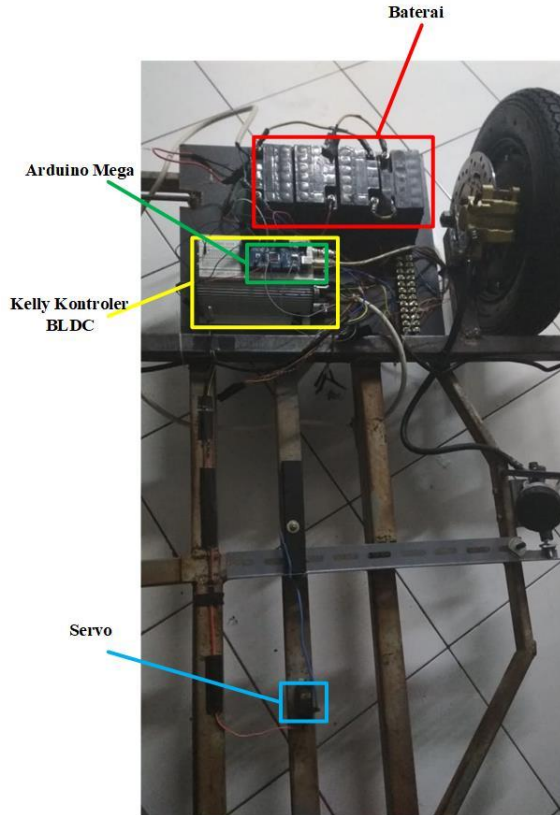


Gambar 3. 6 Servo Towerpro 5010 (Adafruit, n.d.)

Motor servo ini dipilih karena kompatibel dengan kontroler Arduino, tegangan yang diperlukan tidak terlalu besar yaitu sebesar 5V, dan harganya relative lebih murah dibandingkan dengan motor servo dengan torsi yang lebih besar. Servo ini digunakan sebagai aktuator untuk menarik tuas rem cakram. Gambar 3.6 merupakan bentuk fisik dari motor servo yang digunakan.

3.3.6 Hardware Sistem Pengereman

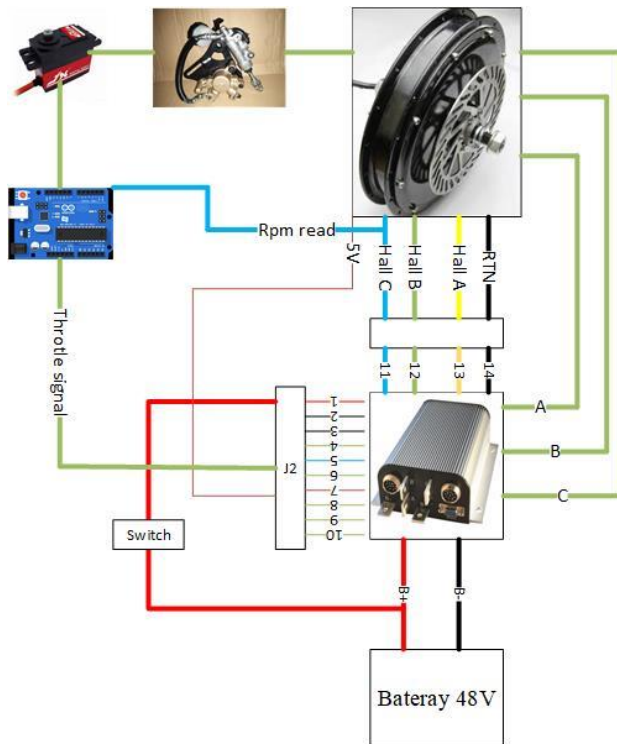
Prototipe mobil dapat dilihat pada gambar 3.7. komponen system penggerak mobil ACePITS terdiri dari baterai, motor BLDC, kontroler BLDC, motor servo dan Arduino mega. Penempatan komponen sistem pengereman ditunjukkan pada gambar 3.7. Sistem pengereman dalam tahap awal prototipe mobil ACePITS menggunakan satu buah rem hidrolik yang diletakan pada roda bagian kanan.hal tersebut dimungkinkan karena motor BLDC dapat melakukan sinkronisasi dalam mengatur kecepatanya.



Gambar 3. 7 Penempatan Komponen

Daya prototipe mobil ACePITS diberikan dari accumulator 12v yang dirangkai secara seri. Yang dihubungkan pada kontroler kelly BLDC. Kontroler Kelly BLDC memiliki pin 3 fase yang terhubung dengan motor BLDC. Dalam prototipe mobil ACePITS ini, pin tersebut dirangkai secara parallel ke dua buah motor BLDC untuk kendali kecepatannya. Dalam pin J2 kontroler BLDC terdapat pin 5 yang digunakan untuk mengatur kecepatan motor, pin

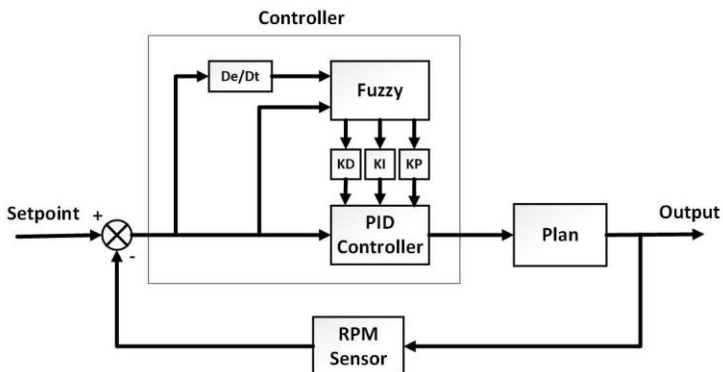
tersebut dihubungkan dengan kontroler Arduino. Pengendalian kecepatan motor dilakukan dengan memberikan output sinyal pengendalian. Pembacaan kecepatan dilakukan dengan memberikan program perhitungan interrupt yang dibaca dari sensor *hall effect*. Pengereman dilakukan dengan motor servo yang dikendalikan oleh kontroler arduino. Dengan memberikan sinyal digital yang berupa posisi dari motor servo. Wiring diagram dari prototipe mobil ACEPITS dapat dilihat pada gambar gambar 3.8.



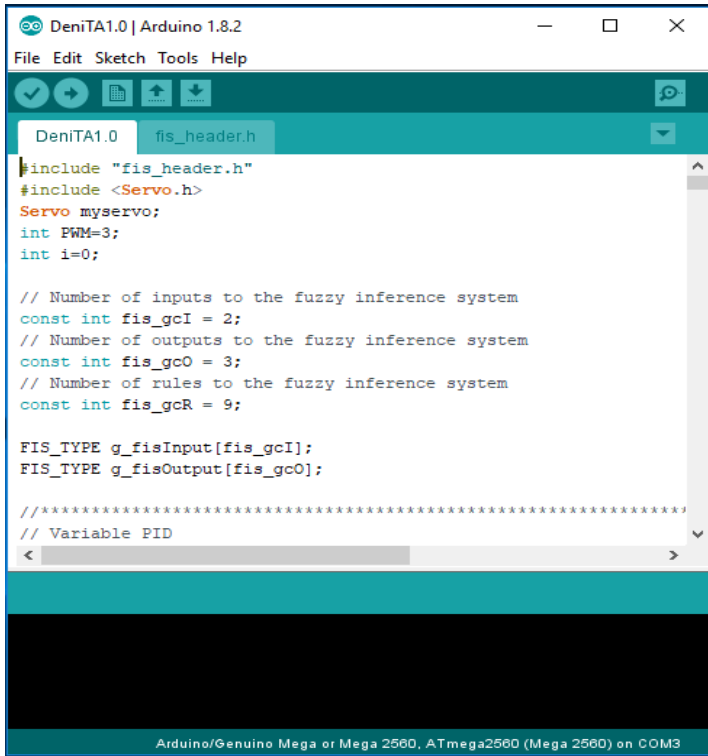
Gambar 3.8 Wiring Sistem Pengereman

3.4 Perancangan Kontroler

Sistem kendali pengereman didesain menggunakan kontroler fuzzy-PID dengan setpoint rpm motor. Sensor *hall effect* pada motor bldc digunakan untuk membaca rpm aktual dari motor. Setelah itu dapat diketahui *error* dari perbedaan antara *setpoint* dengan rpm aktual yang dibaca sensor. Nilai *error* dan perubahan *error* setiap waktu digunakan sebagai input dari sistem fuzzy. Fuzzy inference engine dari sistem fuzzy akan merubah input menjadi output berupa parameter gain berdasarkan rule base fuzzy. Penjelasan mengenai sistem fuzzy akan dibahas pada sub bab selanjutnya. Parameter pengendalian digunakan untuk melakukan *tunning* kendali PID. Kontroler PID memberikan output sinyal pengendalian yang digunakan untuk mengendalikan motor servo. Diagram blok sistem kendali pengeraman ditunjukkan pada gambar 3.9. Perancangan program kontroler dibuat menggunakan Arduino IDE. Sistem kendali pengeraman dibuat menggunakan Bahasa pemrograman C. Kemudian ditanamkan pada kontroler arduino mega.



Gambar 3. 9 Diagram Block Sistem Pengendalian



```

DeniTA1.0 | Arduino 1.8.2
File Edit Sketch Tools Help
DeniTA1.0 fis_header.h
#include "fis_header.h"
#include <Servo.h>
Servo myservo;
int PWM=3;
int i=0;

// Number of inputs to the fuzzy inference system
const int fis_gcI = 2;
// Number of outputs to the fuzzy inference system
const int fis_gcO = 3;
// Number of rules to the fuzzy inference system
const int fis_gcR = 9;

FIS_TYPE g_fisInput[fis_gcI];
FIS_TYPE g_fisOutput[fis_gcO];

//*****
// Variable PID

```

Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) on COM3

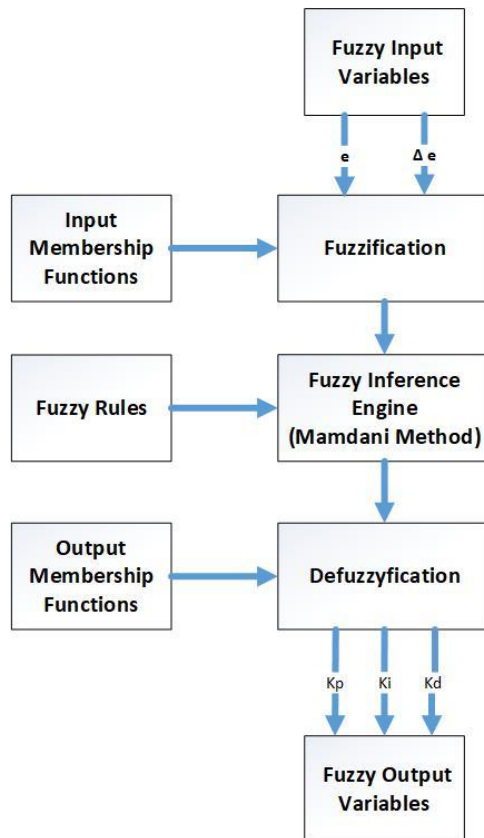
Gambar 3. 10 *Interface software Arduino IDE*

Setpoint rpm ditentukan untuk masukan kontroler sebagai parameter pengendali. Gambar 3.10 menunjukan interface dari *software* Arduino IDE. *Software* tersebut digunakan untuk melakukan *compile* program dari system kendali pengereman yang dibuat. Program kendali pengereman otomatis dapat dilihat pada lampiran.

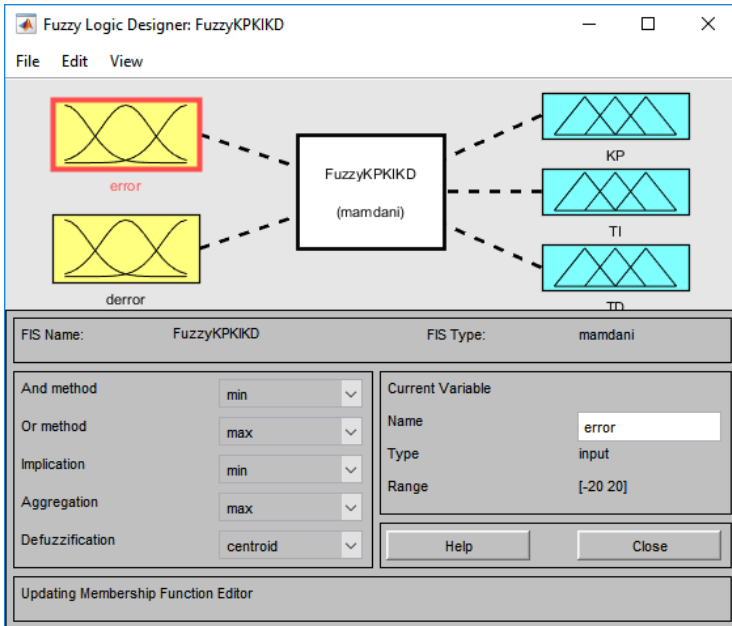
3.5 Perancangan Logika Fuzzy pada Matlab

Perancangan logika fuzzy berupa pembuatan *Fuzzy Inference System* (FIS) pada *software* Matlab. Tipe Fuzzy yang digunakan pada penelitian ini adalah metode Mamdani.

Gambar 3.11 menunjukkan diagram blok perancangan logika fuzzy. Logika fuzzy yang dibuat pada penelitian ini menggunakan 2 input dan 3 output. Input yang berupa bilangan numerik (*crisp*) akan di fuzzifikasi atau dirubah ke dalam himpunan fuzzy (*membership function*). Dari input yang diberikan, kemudian dilakukan konversi menjadi output fuzzy berdasarkan rule base yang telah dibuat. Output yang berupa *membership function* akan didefusifikasi menjadi nilai numerik.



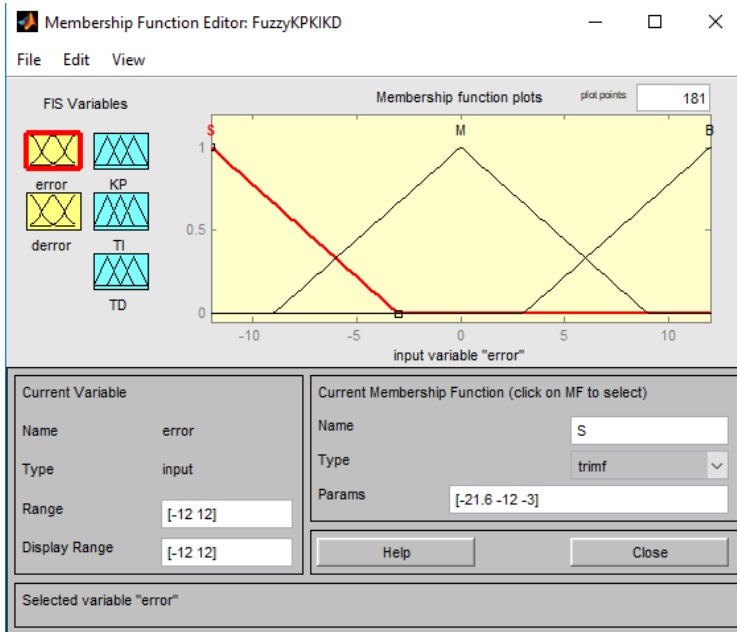
Gambar 3. 11 Diagram Blok Logika Fuzzy



Gambar 3.12 FIS untuk Pengereman

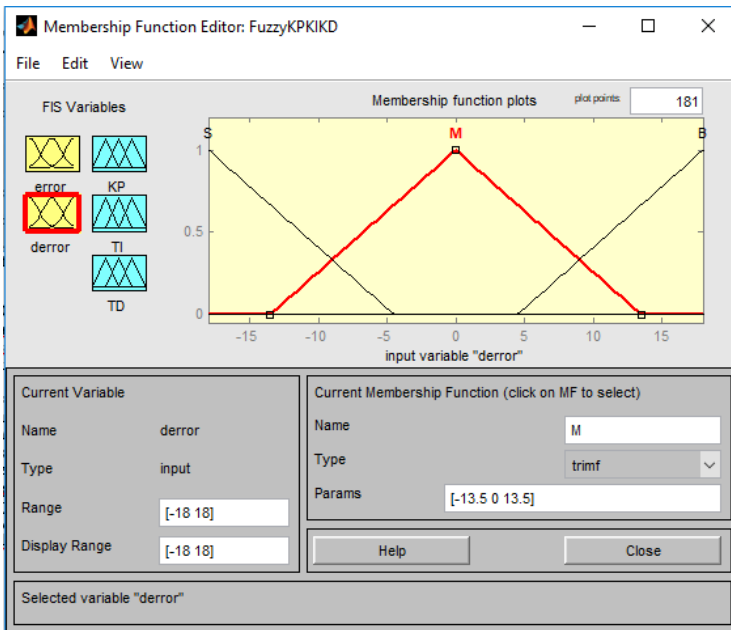
Dapat dilihat pada Gambar 3.10, terdapat dua masukan yaitu *error* dan *delta error*. *Error* adalah selisih keluaran yang dihasilkan dari *setpoint*, *delta error* adalah selisih dari setiap *error* yang terjadi, sedangkan keluarannya berupa 3 parameter kendali PID, yaitu *Kp*, *Ki*, dan *Kd*.

Pada sistem ini, *Fuzzy Inference System* digunakan sebagai sistem pengambilan keputusan untuk menentukan parameter *Kp*, *Ki*, dan *Kd* yang digunakan pada kontroler PID dengan inputan berupa *error* dan selisih setiap *error* pada sistem. Perancangan *fuzzy inference system* dilakukan melalui beberapa tahap, yaitu penentuan fungsi keanggotaan masing-masing *input* dan *output* dari *fuzzy inference system* dan kemudian penentuan aturan *rule base* yang akan menjadi prinsip kerja dari sistem fuzzy yang dirancang.



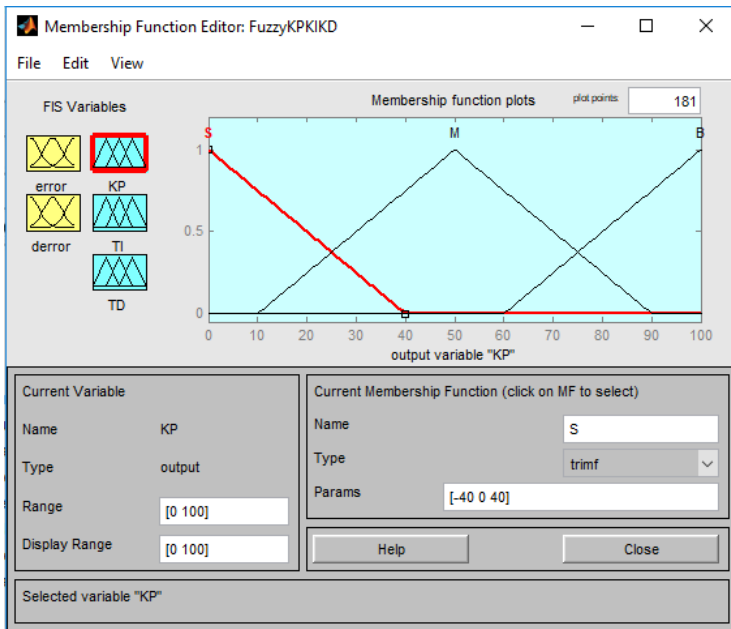
Gambar 3. 13 Fungsi Keanggotaan *Input Error*

Input error dibedakan menjadi 3 buah fungsi keanggotaan yaitu S yang menyatakan nilai error kecil. M bernilai medium atau sedang, dan B yang berarti nilai input error besar. Nilai batas input error bernilai antara -12 sampai 12 yang didapatkan dari penelitian sebelumnya tentang pengendalian sistem rem menggunakan Fuzzy PID. Dalam penelitian ini nilai S menggunakan fungsi keanggotaan segitiga dengan batasan -3, -12. Nilai M menggunakan fungsi keanggotaan segitiga dengan batasan -9, 0, 9. Nilai B menggunakan fungsi keanggotaan segitiga dengan batasan 3, 12. Dengan penentuan batas masing fungsi keanggotaan dilakukan berdasarkan *trial and error*. Gambar 3.13 menunjukkan bentuk fungsi keanggotaan input error.



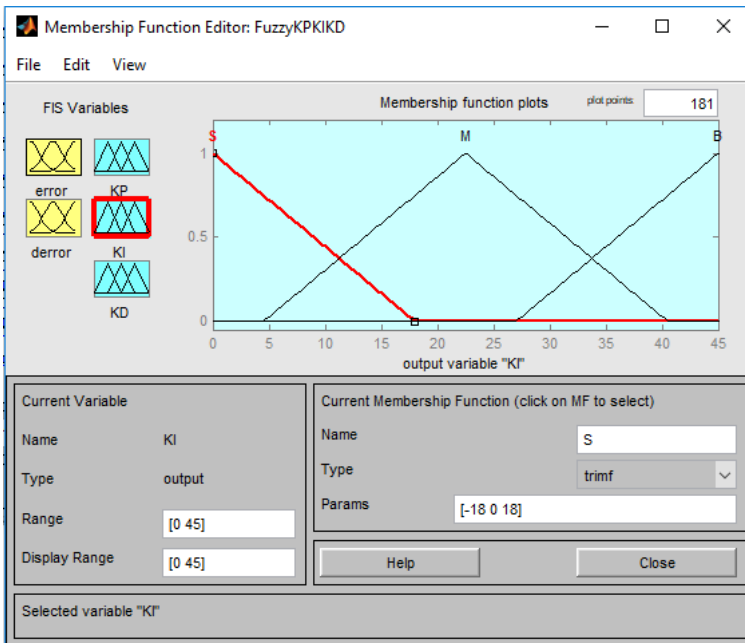
Gambar 3. 14 Fungsi Keanggotaan *Input Delta Error*

Input delta error dibedakan menjadi 3 buah fungsi keanggotaan yaitu S yang menyatakan nilai error kecil. M bernilai medium atau sedang, dan B yang berarti nilai input error besar. Nilai batas input error bernilai antara -18 sampai 18 yang didapatkan dari penelitian sebelumnya yang serupa. Dalam penelitian ini nilai S menggunakan fungsi keanggotaan segitiga dengan batasan - 4,5 , -18. Nilai M menggunakan fungsi keanggotaan segitiga dengan batasan -13,5 ,0 , 13,5. Nilai B menggunakan fungsi keanggotaan segitiga dengan batasan 4,5 , 18. Dengan penentuan batas masing masing fungsi keanggotaan dilakukan berdasarkan *trial and error*. Gambar 3.14 menunjukkan bentuk fungsi keanggotaan input error.



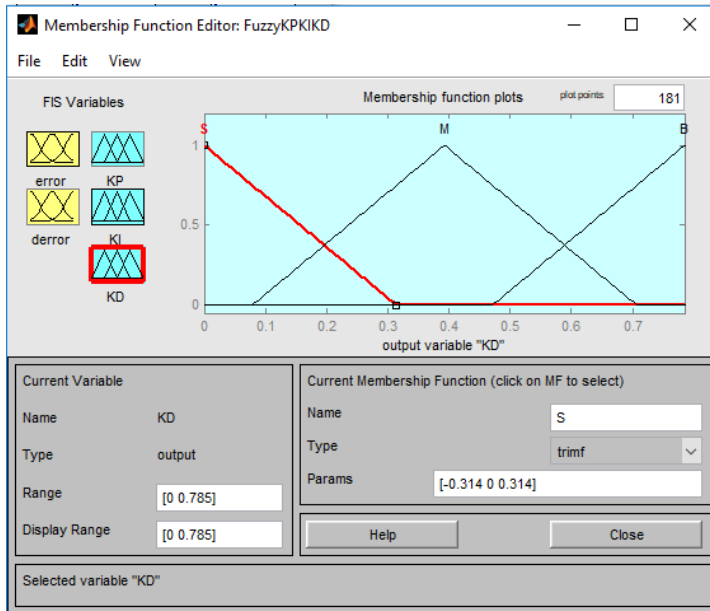
Gambar 3.15 Fungsi Keanggotaan *Output K_p*

Output dari fuzzy nilai K_p dibuat ke dalam 3 buah fungsi keanggotaan yaitu S yang menyatakan nilai error kecil. M bernilai medium atau sedang, dan B yang berarti nilai input error besar. Nilai batas input error bernilai antara 0 sampai 100. Dengan penentuan batas masing masing fungsi keanggotaan dilakukan berdasarkan *trial and error*. Dalam penelitian ini nilai S menggunakan fungsi keanggotaan segitiga dengan batasan 0 , 40. Nilai M menggunakan fungsi keanggotaan segitiga dengan batasan 10 ,50 , 90. Nilai B menggunakan fungsi keanggotaan segitiga dengan batasan 60 , 100. Gambar 3.15 menunjukkan bentuk fungsi keanggotaan output K_p .



Gambar 3.16 Fungsi Keanggotaan Output K_i

Output K_i dibuat ke dalam 3 buah fungsi keanggotaan yaitu S yang menyatakan nilai error kecil, M bernilai medium atau sedang, dan B yang berarti nilai input error besar. Nilai batas input error bernilai antara 0 sampai 45. Dengan penentuan batas masing masing fungsi keanggotaan dilakukan berdasarkan *trial and error*. Dalam penelitian ini nilai S menggunakan fungsi keanggotaan segitiga dengan batasan 0 , 18. Nilai M menggunakan fungsi keanggotaan segitiga dengan batasan 4,5 , 22,5 , 45. Nilai B menggunakan fungsi keanggotaan segitiga dengan batasan 27,5 , 100 Gambar 3.16 menunjukkan bentuk fungsi keanggotaan output K_i .



Gambar 3. 17 Fungsi Keanggotaan Output K_d

Output dari fuzzy nilai K_d dibuat ke dalam 3 buah fungsi keanggotaan yaitu S yang menyatakan nilai error kecil. M bernilai medium atau sedang, dan B yang berarti nilai input error besar. Nilai batas input error bernilai antara 0 sampai 0,785. Dengan penentuan batas masing masing fungsi keanggotaan dilakukan berdasarkan *trial and error*. Dalam penelitian ini nilai S menggunakan fungsi keanggotaan segitiga dengan batasan 0 , 0,325. Nilai M menggunakan fungsi keanggotaan segitiga dengan batasan 0,09 , 0,425 , 0,725. Nilai B menggunakan fungsi keanggotaan segitiga dengan batasan 0,475 , 0,85 Gambar 3.15 menunjukkan bentuk fungsi keanggotaan output K_p .

- **Inference Engine**

Bilangan *crisp* yang dinyatakan dalam fungsi keanggotaan akan diolah dalam *inference engine* yang memiliki prinsip sebab akibat (IF...THEN) di dalam *inference engine* terdapat *rule base* yang mengatur keluaran *fuzzy* sesuai masukan yang diterima sistem. *Rule base* merupakan sekumpulan aturan sebab akibat yang digunakan untuk mengambil keputusan akhir. Dasar perancangan *rule base* pada logika *fuzzy* adalah *trial and error*. Dari input *fuzzy* yang ada (*error* dan *delta error*) dan keluaran parameter *gain* K_p dan K_i dapat dirancang *rule base* sebanyak sembilan buah *rule* dengan logika seperti pada Tabel 3.1.

Table 3. 1 *Rule Base* FIS terhadap Output

	Error/Delta	S	ZO	B
	Error			
Kp	S	S	S	S
	Z	Z	Z	Z
	B	B	B	B
Ki	S	S	S	S
	Z	Z	Z	Z
	B	B	B	B
Kd	S	S	S	S
	Z	Z	Z	Z
	B	B	B	B

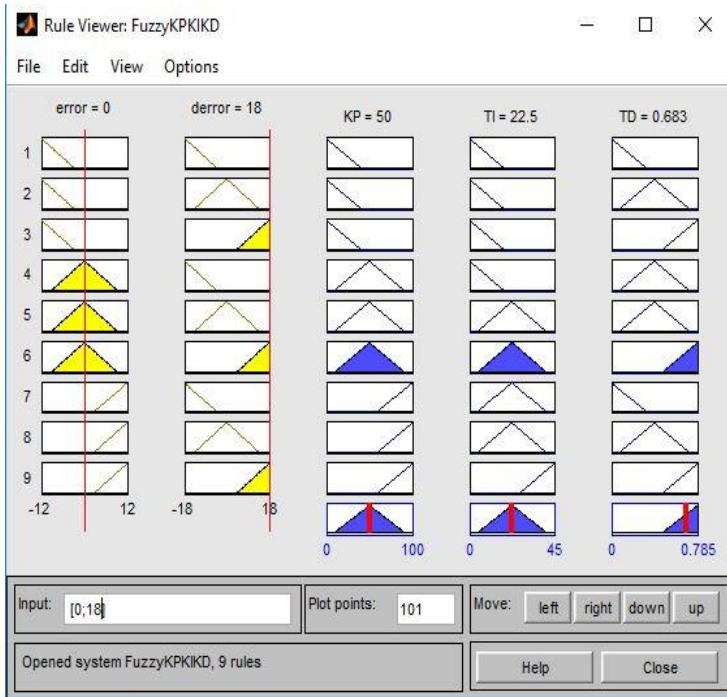
BAB IV ANALISA DATA DAN PEMBAHASAN

4.1 Pengujian Logika Fuzzy

Pengujian logika fuzzy dilakukan menggunakan *software* Matlab untuk mengetahui keluaran dari sistem fuzzy yang dibuat. Pengujian dilakukan dengan memberikan input berupa *error* dan *delta error*. Gambar 4.1 menunjukan pengujian logika fuzzy yang diberikan input *error* dan *delta error* berturut turut 0 dan 0. Hasil dari pengujian didapatkan nilai $K_p = 50$, $K_i = 22.5$ dan $K_d = 0.39$. Dari 9 rule base yang dibuat, pada saat kedua input bernilai 0.



Gambar 4.1 Input *error* = 0 *delta error* =0



Gambar 4.2 Input $error = 0$ delta $error = 18$

Kemudian diberikan *delta error* sebesar 18, dapat dilihat pada Gambar 4.2 nilai K_p dan K_i bernilai tetap sedangkan nilai K_d meningkat menjadi 0.68. Hal ini bertujuan untuk meningkatkan stabilitas keluaran kontroler. Serta mengurangi osilasi karena perubahan *error* naik yang cukup besar.

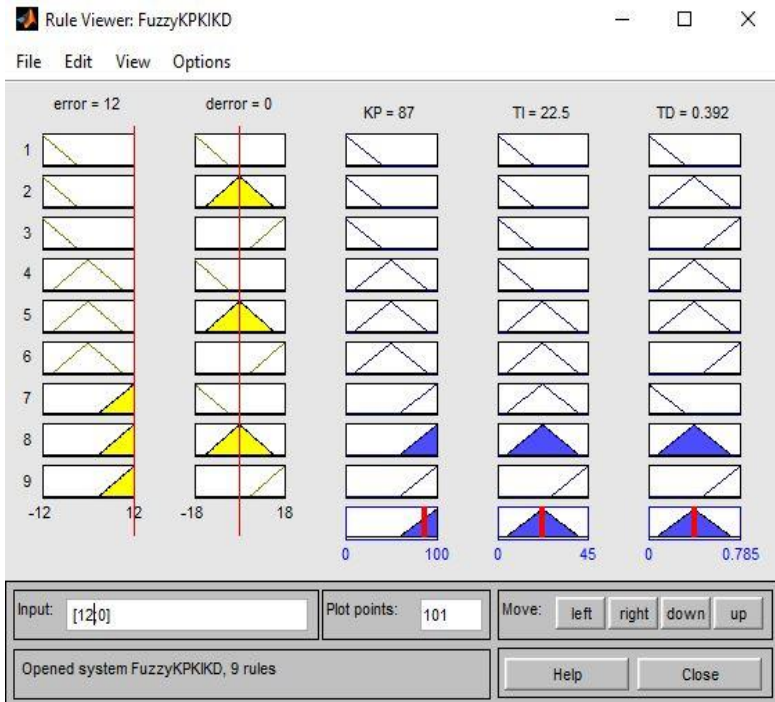
Pada saat terjadi perubahan *error* bernilai negatif (dari respon lebih tinggi dari *setpoint* turun ke nol) seperti pada Gambar 4.3. nilai K_i diperkecil sedangkan nilai K_p dan K_d tetap, hal ini untuk menurunkan kestabilan respon sehingga perubahan *error* diperkecil.



Gambar 4.3 Input $error = 0$ $delta\ error = -18$

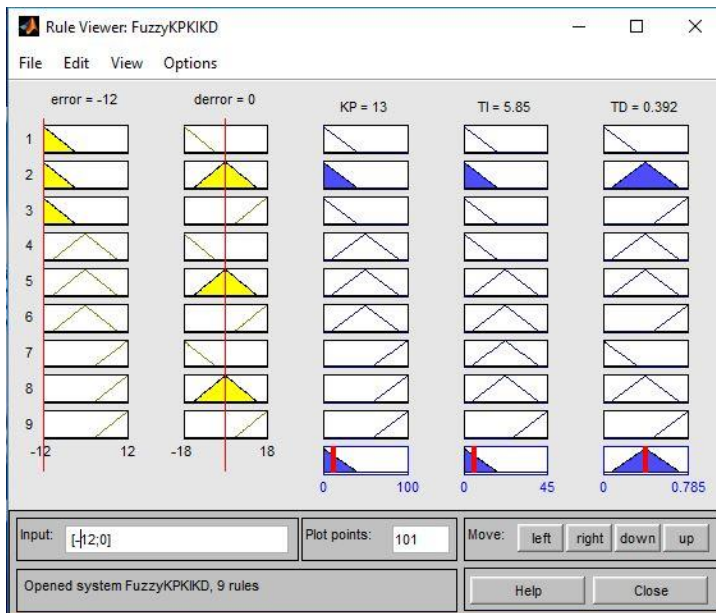
Kondisi pada saat nilai $error$ besar dan perubahan $error$ bernilai nol, nilai gain K_p diperbesar seperti terlihat pada Gambar 4.4. Hal ini bertujuan untuk menurunkan kestabilan sistem sehingga mempercepat respon menuju setpoint.

Pada pengujian sistem fuzzy, dengan pemberian $error$ bernilai -12 dan $delta\ error$ bernilai nol didapatkan nilai K_p , K_i dan K_d berturut-turut 13, 5,85, dan 0,392. Dapat dilihat pada Gambar 4.5 nilai K_p dan K_i diperkecil, hal ini dilakukan untuk memperlambat respon sistem.



Gambar 4.4 Input $error = 12$ delta $error = 0$

Sedangkan pada saat input berada diluar batasan *membership function* nilai K_p , K_i , dan K_d akan ditetapkan dengan nilai berturut turut 50, 22,5 dan 0,392. Pada nilai K_p , K_i dan K_d tersebut respon relatif stabil namun masih terjadi osilasi. Menurut analisa yang saya lakukan, hal tersebut karena pembacaan sensor yang kurang akurat. Pada saat pengujian dengan input erro dan delta error diluar jangkauan input dari sistem fuzzy, output yang dihasilkan berada pada nilai K_p 50, K_i 22,5 dan K_d 0,392.



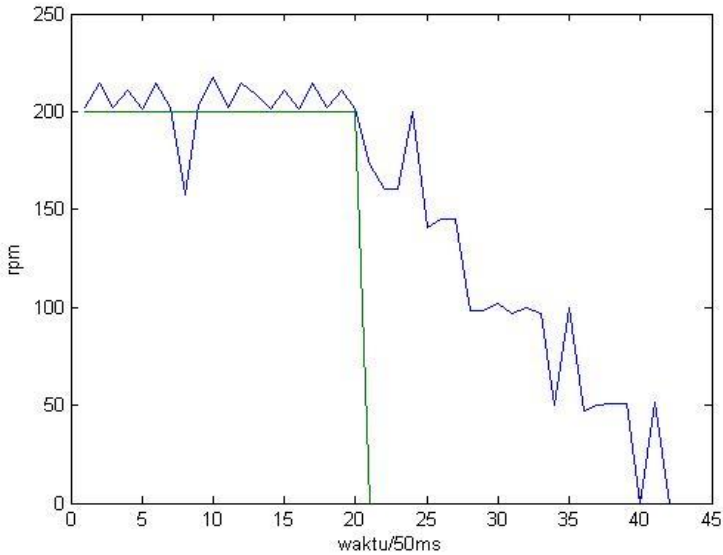
Gambar 4. 5 Input $error$ -12 dan $delta\ error$ 0

4.2 Hasil Pengujian Sistem

Pengujian sistem dilakukan untuk mengetahui karakteristik dari sistem. Hasil dari pengujian kemudian dijadikan acuan untuk membenahi kekurangan dari sistem kendali untuk pengereman otomatis. Pada penelitian ini telah dilakukan uji pengereman tanpa pengereman, uji pengereman menggunakan Fuzzy-PID pada saat mobil pada kecepatan 180, 200 dan 220 rpm.

4.2.1. Hasil Uji Tanpa dilakukan Pengereman

Pengujian ini dilakukan untuk mengetahui performa dan efisiensi dari sistem pengereman sendiri. Pengujian ini dilakukan dengan memberikan setpoint motor sebesar 200 rpm kemudian dilepaskan dan dilakukan pengereman hingga berhenti.

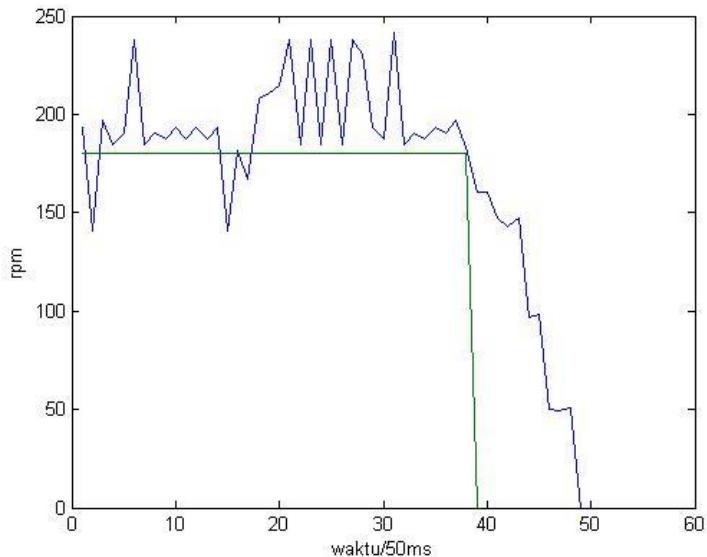


Gambar 4.6 Hasil Uji Tanpa dilakukan Pengereman

Dari hasil uji didapatkan kecepatan awal motor adalah 15 km/jam, waktu yang diperlukan untuk berhenti yaitu 1,1 detik, jarak yang dibutuhkan untuk berhenti adalah 4.6m. Uji tanpa dilakukan pengereman memiliki grafik kecepatan yang kurang stabil dengan osilasi yang cukup banyak terjadi pada saat dilakukan pengereman.

4.2.2. Hasil Uji Pengereman dengan Kecepatan Awal 180rpm

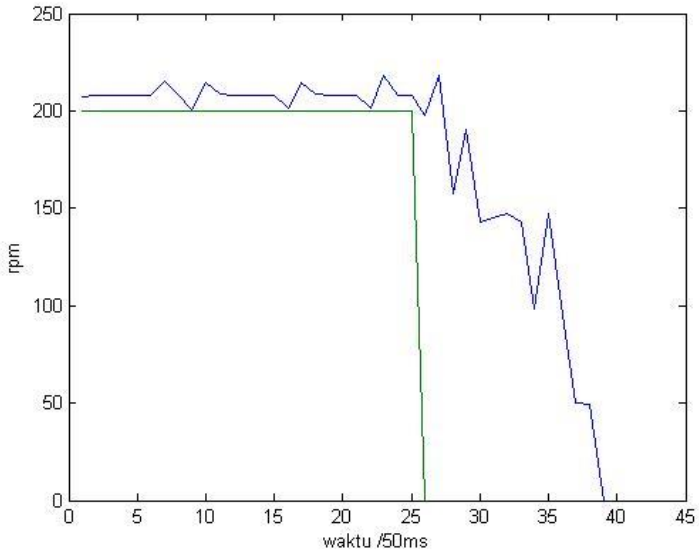
Uji pengereman pada saat setpoint ditetapkan pada 180 rpm dilakukan pengereman hingga motor berhenti. Dari hasil uji didapatkan nilai kecepatan awal 13,5 km/jam, waktu dan jarak yang diperlukan untuk berhenti berturut turut adalah 0.6 detik dan 2,2 meter.



Gambar 4.7 Hasil Uji Pengereman dengan Kecepatan Awal 180rpm

4.2.3. Hasil Uji Pengereman dengan Kecepatan Awal 200 rpm

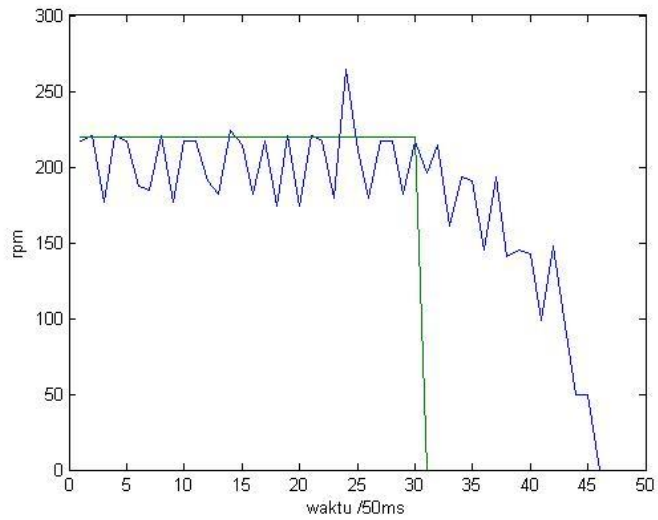
Pengujian ini dilakukan dengan cara memberikan kecepatan awal tertentu pada motor kemudian throtel di lepaskan serta dilakukan pengereman saat itu juga. Dari hasil uji didapatkan kecepatan awal motor adalah 15 km/jam , waktu yang diprlukan untuk berhenti yaitu 0,65 detik, jarak yang dibutuhkan untuk berhenti adalah 2,8 meter. Dari hasil uji pengeraman terjadi penurunan kecepatan yang cukup baik , terlihat pada grafik dengan osilasi yang tidak terlalu besar



Gambar 4.8 Hasil Uji Pengereman dengan Kecepatan Awal 200rpm

4.2.4. Hasil Uji Pengereman dengan Kecepatan Awal 220 rpm

Uji pengereman pada saat setpoint ditetapkan pada 220 rpm dilakukan pengereman hingga motor berhenti. Dari hasil uji didapatkan nilai kecepatan awal 16,5 km/jam, waktu dan jarak yang diperlukan untuk berhenti berturut turut adalah 0,75 detik dan 3,4 meter. Hasil pengujian pada 220 rpm dapat dilihat pada grafik. Osilasi dengan nilai puncak rata rata pada 220rpm, terjadi terjadi karena daya yang diberikan untuk meningkatkan kecepatan motor sudah maksimal.



Gambar 4.9 Hasil Uji Pengereman dengan Kecepatan Awal 220 rpm

Halaman ini sengaja dikosongkan.

BAB V PENUTUP

5.1 Kesimpulan

Adapun kesimpulan yang bisa didapat dari perhitungan yang telah dilakukan adalah sebagai berikut:

- Sistem Pengereman otomatis pada mobil ACePITS dapat dirancang menggunakan sebuah kontroler fuzzy-PID menggunakan rem cakram hidrolis, kontroler arduino dan motor servo. Dari hasil uji pengereman yang dilakukan dengan kecepatan awal 13,5 km/jam, 15 km/jam dan 16,5 km/jam prototipe mobil dapat berhenti pada jarak berturut turut 2,2 meter, 2,8 meter dan 3,4 meter.
- Desain sistem pengereman otomatis dapat dibuat menggunakan fuzzy-PID dengan sistem fuzzy itu sendiri digunakan untuk tuning parameter PID dan kemudian ditanamkan pada kontroler Arduino.

5.2 Saran

Berdasarkan penelitian yang telah dilakukan maka saran yang dapat diberikan untuk penelitian selanjutnya adalah sebagai berikut:

- Metode optimasi tuning seperti genetic algorithm, duelist algorithm, dan particle swarm optimization dapat dilakukan untuk penelitian selanjutnya untuk meningkatkan performa dari sistem kendali pengereman.

Halaman ini sengaja dikosongkan.

Daftar Pustaka

- A. A. Khan, N. R. (2006). Fuzzy PID Controller: Design, Tuning and Comparison with Conventional PID Controller. *IEEE International Conference on Engineering of Intelligent Systems, 1*, 1-6.
- Adafruit. (t.thn.). *Standard Servo - Towerpro SG-5010*. (Adafruit) Dipetik 07 08, 2018, dari <https://www.adafruit.com/product/155>
- Arduino. (2018). *Arduino Mega 2560 REV3*. (Arduino) Dipetik 7 8, 2018, dari <https://store.arduino.cc/usa/arduino-mega-2560-rev3>
- Eneh II, O. P. (2014). Design of an Automatic Brake Control System Using Artificial Neural Network. *International Journal of Scientific & Engineering Research, 5*(4), 1239.
- Gautam, D. (2013). *Control of a Quadrotor Using a Smart Self-Tuning Fuzzy PID Controller*. Korea Selatan: University of Ulsan.
- Hawkins, A. J. (2017, November 7). *Waymo is first to put fully self-driving cars on US roads without a safety driver*. Diambil kembali dari www.theverge.com: <https://www.theverge.com/2017/11/7/16615290/waymo-self-driving-safety-driver-chandler-autonomous>
- Junxia Li, Z. K. (2009). Fuzzy-PID study on new type hydraulic brake system. *International Conference on Intelligent Human-Machine System and Cybernetics, 90-93*.
- K. A Tehrani, A. M. (2012). Introduction to PID Controller - Theory, Tuning and Application to Frontier Areas. *InTech, 10*(5772), 2422.
- K. Astrom, T. H. (1995). *Theory, Design and Tuning. Instrument Society of America*.
- kellycontroller. (2018, 7 10). *HUB Motor 48V1KW(Disc-Brake)*. Diambil kembali dari

- <http://kellycontroller.com>:
<http://kellycontroller.com/hub-motor-48v1kwdisc-brake-p-160.html>
- Litman, T. (2017). Autonomous Vehicle Implementation Predictions. *Victoria Transport Policy Institute*, 1-18.
- Masbi, N. H. (2015). *Rancang Bangun Pengendali Attitude Unmanned Aerial Vehicle*. Surabaya: Teknik Fisika ITS.
- OICA. (2011). *World Motor Vehicle Production By Country and Type*. OICA Correspondents Survey.
- Okezone News. (2018, Januari 08). *Teknologi Mobil yang Dikembangkan di 2018, dari Tenaga Listrik hingga Pendeteksi Kantuk*. Diambil kembali dari news.okezone.com:
<https://news.okezone.com/read/2018/01/08/15/1841823/teknologi-mobil-yang-dikembangkan-di-2018-dari-tenaga-listrik-hingga-pendeteksi-kantuk>
- S, N. N. (2016). *Perancangan Sistem Kendali Attitude pada Pesawat Tanpa Awak Tipe Sayap Tetap dengan Fuzzy PID*. 2016: Jurusan Teknik Fisika ITS.
- SAE International. (2014). *Automated Driving-Levels Of Driving Automation*. Dipetik 7 9, 2018, dari https://web.archive.org/web/20170903105244/https://www.sae.org/misc/pdfs/automated_driving.pdf
- Setiawan, I. (2008). *Kendali PID untuk Proses Industri*. Jakarta: Elex Media Komputindo.
- Sufendi, Trilaksono, B. R., Nasution, S. H., & Purwanto, E. B. (2013). Design and Implementation of Hardware-In-The-Loop-Simulation for UAV Using PID Control Method. *3rd International Conference on Instrumentation, Communications, Information Technology, and Biomedical Engineering (ICICI-BME)*, 124-130.
- TRIBUNNEWS. (2017, Agustus 12). *Beli Mobil Untuk Kebutuhan atau Gengsi Sih*. Diambil kembali dari

TRIBUNNEWS.COM:

<http://www.tribunnews.com/otomotif/2017/08/12/beli-mobil-untuk-kebutuhan-atau-gengsi-sih-baca-tips-berikut-yuk>

WARTA-BPHN. (2015). *Indonesia Merupakan Negara Kepulauan Yang Terbesar Di Dunia*. Jakarta: BPHN.

Wikipedia. (2018, 7 9). *Autonomous Car*. (wikipedia) Dipetik 7 9, 2018, dari https://en.wikipedia.org/wiki/Autonomous_car

Yongbin Ma, Y. L. (2010). *Design of Parameter Self-tuning Fuzzy PID Control for DC Motor*. Hohhot: International Conference on Industrial Mechatronics and Automation.

Zhao, Z. Y., Tomizuka, M., & Isaka, S. (1993). Fuzzy Gain Scheduling of PID Controllers. *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETIC*, 23, 1392-1398.

LAMPIRAN A

```
#include "fis_header.h"
#include <Servo.h>
Servo myservo;
int PWM=3;
int i=0;

// Number of inputs to the fuzzy inference system
const int fis_gcI = 2;
// Number of outputs to the fuzzy inference system
const int fis_gcO = 3;
// Number of rules to the fuzzy inference system
const int fis_gcR = 9;

FIS_TYPE g_fisInput[fis_gcI];
FIS_TYPE g_fisOutput[fis_gcO];

/*****
/ Variable PID
*****/

#include <PID_v1.h>

#define PIN_INPUT 0
#define PIN_OUTPUT 3

//Define Variables we'll be connecting to
double Setpoint, Input, Output;
int error =0;
int derror=0;
//Specify the links and initial tuning parameters
double Kp=2, Ki=5, Kd=1;
PID myPID(&Input, &Output, &Setpoint, Kp, Ki, Kd,
DIRECT);
```

```

//*****
// Variable tachometer
//*****
float rev=0;
float rpm;
int oldtime=0;
int time;

    int potpoin=0;
    int val;

void isr() //interrupt service routine
{
rev++;
}

// Setup routine runs once when you press reset:
void setup()
{
    pinMode(PWM,OUTPUT);
    myservo.attach(9);
    Serial.begin(9600);
    Serial1.begin(9600);

attachInterrupt(digitalPinToInterrupt(2),isr,RISING);//attachin
g the interrupt

//Input = analogRead(PIN_INPUT);
Setpoint = 0;

//turn the PID on
myPID.SetMode(AUTOMATIC);

```

```

// // initialize the Analog pins for input.
// // Pin mode for Input: error
// pinMode(0 , INPUT);
// // Pin mode for Input: derror
// pinMode(1 , INPUT);
//
// // initialize the Analog pins for output.
// // Pin mode for Output: KP
// pinMode(2 , OUTPUT);
// // Pin mode for Output: TI
// pinMode(3 , OUTPUT);
// // Pin mode for Output: TD
// pinMode(4 , OUTPUT);
}

// Loop routine runs over and over again forever:
void loop()
{
  bluetut();
  tacho();
  int error0 = error;
  error = Setpoint-rpm;
  derror= error-error0;

  Serial.print(error);
  Serial.print("\t");
  Serial.print(derror);
  Serial.print("\t");

  // Read Input: error
  g_fisInput[0] = error;
  // Read Input: derror
  g_fisInput[1] = derror;

```

```

g_fisOutput[0] = 0;
g_fisOutput[1] = 0;
g_fisOutput[2] = 0;

fis_evaluate();

Serial.print( g_fisOutput[0]);
Serial.print("\t");
Serial.print( g_fisOutput[1]);
Serial.print("\t");
Serial.print( g_fisOutput[2]);
Serial.print("\t");

Input=error;
myPID.SetTunings(g_fisOutput[0],          g_fisOutput[1],
g_fisOutput[2]);
myPID.Compute();
analogWrite(PIN_OUTPUT, Output);
Serial.print(Output);
Serial.print("\t");

// // Set output vlaue: KP
// analogWrite(2 , g_fisOutput[0]);
// // Set output vlaue: TI
// analogWrite(3 , g_fisOutput[1]);
// // Set output vlaue: TD
// analogWrite(4 , g_fisOutput[2]);

val= Output;
val= map(val,0,255,0,180);

Serial.print(val);

```

```

Serial.print("\t");
Serial.print(Setpoint);
Serial.print("\t");
Serial.println(i);

myservo.write(val);
//delay(15);
}

void bluetut(){

while(Serial1.available())
{
char blue=Serial1.read();
analogWrite(PWM,i);

if(blue=='1') //mundur
{
i=i+5;

}
else if(blue=='2') //maju
{
i=i-5;
}
else if(blue=='3') //majukanan
{
i=0;
}
else if(blue=='4') //majukiri
{
i=255;
}
else if(blue=='5') //mundurkanan

```

```

    {
        i=0;
        myservo.write(0);

    }
    else if(blue=='6') //mundurkiri
    {
        i=0;
        myservo.write(180);
    }
    else if(blue=='7') //mundurkiri
    {
        Setpoint= Setpoint+10;
    }
    else if(blue=='8') //mundurkiri
    {
        Setpoint= Setpoint-10;
    }

    delay(100);
}

}

//*****
// Function tachometer
//*****

void tacho(){

    delay(50);

```

```

//detachInterrupt(0); //detaches the interrupt //saves
the current time
time=millis()-oldtime; //finds the time
rpm=(rev/time)*3000; //calculates rpm
oldtime=millis();
Serial.print(rpm);
Serial.print("\t");
Serial.print(time);
Serial.print("\t");
Serial.print(rev);
Serial.print("\t");
rev=0;
}

```

```

/*****

```

```

// Support functions for Fuzzy Inference System

```

```

*****/

```

```

// Triangular Member Function

```

```

FIS_TYPE fis_trimf(FIS_TYPE x, FIS_TYPE* p)

```

```

{
    FIS_TYPE a = p[0], b = p[1], c = p[2];
    FIS_TYPE t1 = (x - a) / (b - a);
    FIS_TYPE t2 = (c - x) / (c - b);
    if ((a == b) && (b == c)) return (FIS_TYPE) (x == a);
    if (a == b) return (FIS_TYPE) (t2*(b <= x)*(x <= c));
    if (b == c) return (FIS_TYPE) (t1*(a <= x)*(x <= b));
    t1 = min(t1, t2);
    return (FIS_TYPE) max(t1, 0);
}

```

```

FIS_TYPE fis_min(FIS_TYPE a, FIS_TYPE b)

```

```

{
    return min(a, b);
}

```

```

FIS_TYPE fis_max(FIS_TYPE a, FIS_TYPE b)
{
    return max(a, b);
}

```

```

FIS_TYPE fis_array_operation(FIS_TYPE *array, int size,
_FIS_ARR_OP pfnOp)
{
    int i;
    FIS_TYPE ret = 0;

    if (size == 0) return ret;
    if (size == 1) return array[0];

    ret = array[0];
    for (i = 1; i < size; i++)
    {
        ret = (*pfnOp)(ret, array[i]);
    }

    return ret;
}

```

```

//*****
// Data for Fuzzy Inference System
//*****
// Pointers to the implementations of member functions
_FIS_MF fis_gMF[] =
{
    fis_trimf
};

// Count of member function for each Input

```



```

int fis_gIMFCount[] = { 3, 3 };

// Count of member function for each Output
int fis_gOMFCount[] = { 3, 3, 3 };

// Coefficients for the Input Member Functions
FIS_TYPE fis_gMFI0Coeff1[] = { -50, -20, -5 };
FIS_TYPE fis_gMFI0Coeff2[] = { -15, 0, 15 };
FIS_TYPE fis_gMFI0Coeff3[] = { 5, 20, 60 };
FIS_TYPE* fis_gMFI0Coeff[] = { fis_gMFI0Coeff1,
fis_gMFI0Coeff2, fis_gMFI0Coeff3 };
FIS_TYPE fis_gMFI1Coeff1[] = { -15, 0, 15 };
FIS_TYPE fis_gMFI1Coeff2[] = { -50, -20, -5 };
FIS_TYPE fis_gMFI1Coeff3[] = { 5, 20, 60 };
FIS_TYPE* fis_gMFI1Coeff[] = { fis_gMFI1Coeff1,
fis_gMFI1Coeff2, fis_gMFI1Coeff3 };
FIS_TYPE** fis_gMFICoeff[] = { fis_gMFI0Coeff,
fis_gMFI1Coeff };

// Coefficients for the Output Member Functions
FIS_TYPE fis_gMFO0Coeff1[] = { -12, 0, 12 };
FIS_TYPE fis_gMFO0Coeff2[] = { 3, 15, 27 };
FIS_TYPE fis_gMFO0Coeff3[] = { 18, 30, 42 };
FIS_TYPE* fis_gMFO0Coeff[] = { fis_gMFO0Coeff1,
fis_gMFO0Coeff2, fis_gMFO0Coeff3 };
FIS_TYPE fis_gMFO1Coeff1[] = { -10, 0, 10 };
FIS_TYPE fis_gMFO1Coeff2[] = { 2.5, 12.5, 22.5 };
FIS_TYPE fis_gMFO1Coeff3[] = { 15, 25, 35 };
FIS_TYPE* fis_gMFO1Coeff[] = { fis_gMFO1Coeff1,
fis_gMFO1Coeff2, fis_gMFO1Coeff3 };
FIS_TYPE fis_gMFO2Coeff1[] = { -1.2, 0, 1.2 };
FIS_TYPE fis_gMFO2Coeff2[] = { 0.3, 1.5, 2.7 };
FIS_TYPE fis_gMFO2Coeff3[] = { 1.8, 3, 4.2 };
FIS_TYPE* fis_gMFO2Coeff[] = { fis_gMFO2Coeff1,
fis_gMFO2Coeff2, fis_gMFO2Coeff3 };

```

```
FIS_TYPE** fis_gMFOCoeff[] = { fis_gMFO0Coeff,  
fis_gMFO1Coeff, fis_gMFO2Coeff };
```

```
// Input membership function set  
int fis_gMFI0[] = { 0, 0, 0 };  
int fis_gMFI1[] = { 0, 0, 0 };  
int* fis_gMFI[] = { fis_gMFI0, fis_gMFI1 };
```

```
// Output membership function set  
int fis_gMFO0[] = { 0, 0, 0 };  
int fis_gMFO1[] = { 0, 0, 0 };  
int fis_gMFO2[] = { 0, 0, 0 };  
int* fis_gMFO[] = { fis_gMFO0, fis_gMFO1, fis_gMFO2};
```

```
// Rule Weights  
FIS_TYPE fis_gRWeight[] = { 1, 1, 1, 1, 1, 1, 1, 1, 1 };
```

```
// Rule Type  
int fis_gRType[] = { 1, 1, 1, 1, 1, 1, 1, 1, 1 };
```

```
// Rule Inputs  
int fis_gRI0[] = { 1, 2 };  
int fis_gRI1[] = { 1, 1 };  
int fis_gRI2[] = { 1, 3 };  
int fis_gRI3[] = { 2, 2 };  
int fis_gRI4[] = { 2, 1 };  
int fis_gRI5[] = { 2, 3 };  
int fis_gRI6[] = { 3, 2 };  
int fis_gRI7[] = { 3, 1 };  
int fis_gRI8[] = { 3, 3 };  
int* fis_gRI[] = { fis_gRI0, fis_gRI1, fis_gRI2, fis_gRI3,  
fis_gRI4, fis_gRI5, fis_gRI6, fis_gRI7, fis_gRI8 };
```

```
// Rule Outputs  
int fis_gRO0[] = { 1, 1, 1 };
```

```

int fis_gRO1[] = { 1, 1, 2 };
int fis_gRO2[] = { 1, 1, 3 };
int fis_gRO3[] = { 2, 1, 2 };
int fis_gRO4[] = { 2, 2, 2 };
int fis_gRO5[] = { 2, 2, 3 };
int fis_gRO6[] = { 3, 2, 1 };
int fis_gRO7[] = { 3, 2, 2 };
int fis_gRO8[] = { 3, 3, 3 };
int* fis_gRO[] = { fis_gRO0, fis_gRO1, fis_gRO2, fis_gRO3,
fis_gRO4, fis_gRO5, fis_gRO6, fis_gRO7, fis_gRO8 };

```

```

// Input range Min
FIS_TYPE fis_gIMin[] = { -20, -20 };

```

```

// Input range Max
FIS_TYPE fis_gIMax[] = { 20, 20 };

```

```

// Output range Min
FIS_TYPE fis_gOMin[] = { 0, 0, 0 };

```

```

// Output range Max
FIS_TYPE fis_gOMax[] = { 30, 25, 3 };

```

```

/*****
// Data dependent support functions for Fuzzy Inference
System
*****/
FIS_TYPE    fis_MF_out(FIS_TYPE**    fuzzyRuleSet,
FIS_TYPE x, int o)
{
    FIS_TYPE mfOut;
    int r;

    for (r = 0; r < fis_gcR; ++r)
    {

```

```

    int index = fis_gRO[r][o];
    if (index > 0)
    {
        index = index - 1;
        mfOut = (fis_gMF[fis_gMFO[o][index]])(x,
fis_gMFOCoeff[o][index]);
    }
    else if (index < 0)
    {
        index = -index - 1;
        mfOut = 1 - (fis_gMF[fis_gMFO[o][index]])(x,
fis_gMFOCoeff[o][index]);
    }
    else
    {
        mfOut = 0;
    }

    fuzzyRuleSet[0][r] = fis_min(mfOut,
fuzzyRuleSet[1][r]);
}
return fis_array_operation(fuzzyRuleSet[0], fis_gcR,
fis_max);
}

```

```

FIS_TYPE fis_defuzz_centroid(FIS_TYPE** fuzzyRuleSet,
int o)
{
    FIS_TYPE step = (fis_gOMax[o] - fis_gOMin[o]) /
(FIS_RESOLUTION - 1);
    FIS_TYPE area = 0;
    FIS_TYPE momentum = 0;
    FIS_TYPE dist, slice;
    int i;

```

```

// calculate the area under the curve formed by the MF
outputs
for (i = 0; i < FIS_RESOLUTION; ++i){
    dist = fis_gOMin[o] + (step * i);
    slice = step * fis_MF_out(fuzzyRuleSet, dist, o);
    area += slice;
    momentum += slice*dist;
}

return ((area == 0) ? ((fis_gOMax[o] + fis_gOMin[o]) / 2) :
(momentum / area));
}

//*****
// Fuzzy Inference System
//*****
void fis_evaluate()
{
    FIS_TYPE fuzzyInput0[] = { 0, 0, 0 };
    FIS_TYPE fuzzyInput1[] = { 0, 0, 0 };
    FIS_TYPE* fuzzyInput[fis_gcI] = { fuzzyInput0,
fuzzyInput1, };
    FIS_TYPE fuzzyOutput0[] = { 0, 0, 0 };
    FIS_TYPE fuzzyOutput1[] = { 0, 0, 0 };
    FIS_TYPE fuzzyOutput2[] = { 0, 0, 0 };
    FIS_TYPE* fuzzyOutput[fis_gcO] = { fuzzyOutput0,
fuzzyOutput1, fuzzyOutput2, };
    FIS_TYPE fuzzyRules[fis_gcR] = { 0 };
    FIS_TYPE fuzzyFires[fis_gcR] = { 0 };
    FIS_TYPE* fuzzyRuleSet[] = { fuzzyRules, fuzzyFires };
    FIS_TYPE sW = 0;

// Transforming input to fuzzy Input
int i, j, r, o;
for (i = 0; i < fis_gcI; ++i)

```

```

{
    for (j = 0; j < fis_gIMFCount[i]; ++j)
    {
        fuzzyInput[i][j] =
            (fis_gMF[fis_gMFI[i][j]])(g_fisInput[i],
fis_gMFICoeff[i][j]);
    }
}

int index = 0;
for (r = 0; r < fis_gcR; ++r)
{
    if (fis_gRType[r] == 1)
    {
        fuzzyFires[r] = FIS_MAX;
        for (i = 0; i < fis_gcI; ++i)
        {
            index = fis_gRI[r][i];
            if (index > 0)
                fuzzyFires[r] = fis_min(fuzzyFires[r],
fuzzyInput[i][index - 1]);
            else if (index < 0)
                fuzzyFires[r] = fis_min(fuzzyFires[r], 1 -
fuzzyInput[i][-index - 1]);
            else
                fuzzyFires[r] = fis_min(fuzzyFires[r], 1);
        }
    }
    else
    {
        fuzzyFires[r] = FIS_MIN;
        for (i = 0; i < fis_gcI; ++i)
        {
            index = fis_gRI[r][i];
            if (index > 0)

```

```

        fuzzyFires[r] = fis_max(fuzzyFires[r],
fuzzyInput[i][index - 1]);
        else if (index < 0)
            fuzzyFires[r] = fis_max(fuzzyFires[r], 1 -
fuzzyInput[i][-index - 1]);
        else
            fuzzyFires[r] = fis_max(fuzzyFires[r], 0);
    }
}

fuzzyFires[r] = fis_gRWeight[r] * fuzzyFires[r];
sW += fuzzyFires[r];
}

if (sW == 0)
{
    for (o = 0; o < fis_gcO; ++o)
    {
        g_fisOutput[o] = ((fis_gOMax[o] + fis_gOMin[o]) / 2);
    }
}
else
{
    for (o = 0; o < fis_gcO; ++o)
    {
        g_fisOutput[o] = fis_defuzz_centroid(fuzzyRuleSet,
o);
    }
}
}
}

```


BIOGRAFI PENULIS



Penulis dengan nama lengkap Deni Agprianta dilahirkan di Klaten pada tanggal 09 Mei 1996 dari ayah yang bernama Agus Wiranta dan Ibu bernama Supriyanti. Penulis telah menyelesaikan pendidikan formal di SDN 2 Padas (2002-2008), SMPN 1 Karangnom Klaten (2008-2011), SMA Negeri 1 Karangnom (2011-2014). Setelah menyelesaikan pendidikan SMA, penulis diterima di Jurusan Teknik Fisika FTI ITS Surabaya melalui jalur SNMPTN dengan Program Studi S1 Reguler Teknik Fisika dan terdaftar dengan NRP 2414100006. Selama menempuh pendidikan di Teknik Fisika ITS penulis pernah melakukan Kerja Praktek di PT Dirgantara Indonesia pada Agustus 2017. Penulis mengambil bidang minat Rekayasa Instrumentasi dan Kontrol dalam menyelesaikan Tugas Akhir jenjang S1 dibawah bimbingan Andi Rahmadiansah, ST, MT dengan penelitian yang berjudul **“RANCANG BANGUN SISTEM PENEREMAN OTOMATIS BERBASIS FUZZY PID PADA PROTOTYPE MOBIL OTONOM ACePITS.”** Bagi pembaca yang memiliki kritik, saran, atau ingin berdiskusi lebih lanjut mengenai Tugas Akhir ini maka dapat menghubungi penulis melalui email deniagprianta@gmail.com.