



**SISTEM KONTROL KECEPATAN DAN *MONITORING*
MOBIL LISTRIK BERBASIS LOGIKA *FUZZY* DAN
*INTERNET OF THINGS***

SKRIPSI

Oleh
Hendrik Iswanto
NIM 161910201101

**PROGRAM STUDI STRATA 1 TEKNIK ELEKTRO
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS JEMBER
2020**



**SISTEM KONTROL KECEPATAN DAN *MONITORING*
MOBIL LISTRIK BERBASIS LOGIKA *FUZZY* DAN
*INTERNET OF THINGS***

SKRIPSI

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat
untuk menyelesaikan Program Studi Teknik Elektro (S1)
dan mencapai gelar Sarjana Teknik

Oleh
Hendrik Iswanto
NIM 161910201101

**PROGRAM STUDI STRATA 1 TEKNIK ELEKTRO
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS JEMBER
2020**

PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

1. Allah SWT, atas segala limpahan rahmat nikmat, hidayah, dan inayah-Nya sehingga berkesempatan menimba ilmu di Fakultas Teknik Universitas Jember;
2. Rasulullah Muhammad SAW, sebagai *uswah* terbaik sepanjang masa;
3. Kedua orang tua saya Bapak Sumarno dan Ibu Ida yang telah memberikan ketulusan doa, kasih sayang, motivasi, dukungan, dan semangat yang luar biasa;
4. Bapak Maman Nur Jaman dan Ibu Karminah yang telah memberikan ketulusan doa, kasih sayang, motivasi, dukungan, dan semangat yang luar biasa;
5. Mbak Munawaroh dan Mbak Eka Yatur Rohmah yang telah memberikan ketulusan doa, kasih sayang, motivasi, dukungan, dan semangat yang luar biasa;
6. Mas Ervan, Mas Adi, Mas Putra, dan adik-adikku Alfira, Dian, Ari, Diva yang telah menghibur dan memberikan semangat yang luar biasa;
7. Seluruh Dosen Teknik Elektro yang telah memberikan bekal ilmu pengetahuan dan bimbingan selama mengikuti pendidikan di Jurusan Teknik Elektro Fakultas Teknik universitas Jember;
8. Guru- guru saya sejak taman kanak-kanak sampai perguruan tinggi yang telah memberikan bekal ilmu pengetahuan dan bimbingan;
9. Teman-teman Teknik Elektro Universitas Jember angkatan 2016 “INDUKTRO” yang sangat membantu, motivasi, dukungan, semangat dan menemani perjalanan masa perkuliahan yang sudah seperti keluarga dari awal kuliah;
10. Tim Riset Mobil Listrik TITEN Universitas Jember yang telah memberi pengalaman yang sangat luar biasa bagi saya yang tak bisa saya lupakan;
11. Teman-teman diskusi saya yang telah memberi arahan dan saran yang sangat membangun Andri, Thofiq, Andre, Anam, Kholid, Tanu, Dimas, Agung,

Guntur, dan Tatit yang telah menjadi teman seperjuangan semoga selalu diberikan kesehatan dan kesuksesan;

12. Teman-teman BPUN 2016 Jember Andri, Adiet, Bangkit, Puspa dan Nanda yang telah menjadi teman seperjuangan semoga selalu diberikan kesehatan dan kesuksesan;
13. Teman satu kontrakan Andri, Najemi, Dani, Alfian, dan Fahrul yang telah menjadi teman suka maupun duka;
14. Teman-teman paguyuban FORSMAWI Jember yang telah menjadi keluarga saya di tanah perantauan;
15. Teman-teman UKM-O ESTER Fakultas Teknik Universitas Jember yang telah mengajarkan bagaimana berkehidupan dengan orang yang beraneka ragam latar belakang;
16. Teman-teman KKN-34 Kalibendo Periode 1 2020 yang telah menjadi teman suka maupun duka;

MOTTO

Katakanlah, “Wahai hamba-hamba-Ku yang melampaui batas terhadap diri mereka sendiri! Janganlah kamu berputus asa dari rahmat Allah. Sesungguhnya Allah mengampuni dosa-dosa semuanya. Sungguh, Dialah Yang Maha Pengampun, Maha Penyayang.
(terjemahan Surat *Az Zumar* ayat 53)^{*)}

Sesungguhnya sesudah kesulitan itu ada kemudahan. Maka apabila telah selesai (dari suatu urusan), kerjakanlah dengan sungguh-sungguh (urusan) yang lain. Dan hanya kepada Tuhanmulah hendaknya kamu berharap.
(terjemahan Surat *Al Insyirah* ayat 6-8)^{**)}

You are responsible for your life. You can't keep blaming somebody else for your dysfunction. Life is really about moving on.^{***)}

^{*)}Qur'an Kemenag: *Lajnah Pentashihan Mushaf Al-Qur'an Buku & Referensi*
<https://quran.kemenag.go.id>, diakses pada 8 November 2020 pukul 11:30

^{**)}Kementerian Agama Republik Indonesia. 2014. *Al-Qur'anulkarim & Terjemah*. Surakarta: Ziyad Books

^{***)}Oprah Winfrey. 2014. *What I Know For Sure*. United State of America: Flatiron Books.

PERNYATAAN

Saya yang bertanda tangan dibawah ini:

Nama : Hendrik Iswanto

NIM : 161910201101

Menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Sistem Kontrol Kecepatan dan *Monitoring* Mobil Listrik Berbasis Logika *Fuzzy* dan *Internet of Things*” adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada institusi mana pun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa ada tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Jember, 10 November 2020

Yang menyatakan,

Hendrik Iswanto

NIM 161910201101

SKRIPSI

**SISTEM KONTROL KECEPATAN DAN *MONITORING*
MOBIL LISTRIK BERBASIS LOGIKA *FUZZY* DAN
*INTERNET OF THINGS***

oleh
Hendrik Iswanto
NIM 161910201101

Pembimbing

Dosen Pembimbing Utama : Prof. Dr. Ir. Bambang Sujanarko, M.M.

Dosen Pembimbing Anggota : Dr. Triwahju Hardianto, S.T., M.T.

PENGESAHAN

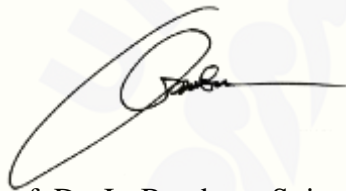
Skripsi berjudul “Sistem Kontrol Kecepatan dan *Monitoring* Mobil Listrik Berbasis Logika *Fuzzy* dan *Internet of Things*” karya Hendrik Iswanto telah diuji dan disahkan pada:

hari, tanggal : Rabu, 04 November 2020

tempat : Fakultas Teknik Universitas Jember.

Tim Penguji:

Ketua,



Prof. Dr. Ir. Bambang Sujanarko, M.M.
NIP 196312011994021002

Anggota I,



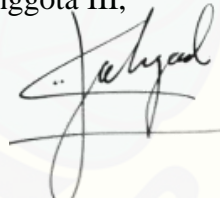
Dr. Triwahju Hardianto, S.T., M.T.
NIP 197008261997021001

Anggota II,



Ir. Widyono Hadi, M.T.
NIP 196104141989021001

Anggota III,



Widya Cahyadi, S.T., M.T.
NIP 198511102014041001

Mengesahkan

Dekan Fakultas Teknik,

Dr. Triwahju Hardianto, S.T., M.T.

NIP 197008261997021001

RINGKASAN

Sistem Kontrol Kecepatan dan *Monitoring* Mobil Listrik Berbasis Logika Fuzzy dan *Internet of Things*; Hendrik Iswanto; 161910201101; 2020; 121 halaman; Jurusan Teknik Elektro, Fakultas Teknik, Universitas Jember.

Perkembangan teknologi baru perlu dikembangkan untuk melakukan penghematan energi fosil maupun untuk penerapan sumber energi baru, salah satunya di bidang transportasi yang terus berkembang yaitu kendaraan ramah lingkungan. Hal ini dipicu oleh meningkatnya emisi gas rumah kaca dunia seiring pertumbuhan populasi dan konsumsi bahan bakar fosil. Kendaraan listrik menjadi salah satu solusi dalam pemecahan masalah yang dihadapi oleh sektor otomotif di seluruh dunia untuk kelestarian lingkungan dan energi. Perkembangan industri menjadi tantangan bagi para insinyur untuk terlibat langsung dalam merancang perangkat yang bisa diimplementasikan di sektor industri. *Internet of Things* (IoT) merupakan sebuah inovasi yang dapat mengumpulkan dan mengirim data sensor melalui internet yang disimpan di *cloud*.

Penggunaan teknologi *Internet of Things* (IoT) pada sistem *monitoring* mobil listrik TITEN EV-3 dapat dilakukan melalui Blynk di mana pun secara *online* dan *real time*. Aplikasi Blynk digunakan untuk *monitoring* tegangan, arus, daya, energi, sudut, pedal gas, PWM, kecepatan, dan jarak tempuh mobil listrik. Sistem FLC digunakan manajemen kinerja kontrol motor BLDC saat di tanjakan dan turunan dengan tipe Mamdani. Variabel *input* berupa nilai ADC potensiometer dan sudut dari sensor MPU 6050, sedangkan variabel *output* berupa PWM dari Arduino DUE. Kecepatan mobil listrik dikendalikan secara otomatis dengan menambahkan atau menurunkan PWM kontrol BLDC menggunakan algoritma *fuzzy logic controller* sesuai *rule-base*.

Penambahan atau pengurangan PWM dikendalikan sistem FLC berdasarkan variabel *input* POT dan sudut. Pada penelitian dengan sudut kemiringan jalan sebesar $18,61^\circ$ sehingga diketahui *grade resistance* kendaraan saat di tanjakan dan

turunan sebesar 625,16 N. Nilai arus terbesar pada pengujian sistem *monitoring* melalui aplikasi Blynk saat mobil listrik berjalan menanjak yaitu sebesar 14,21 A. Hal ini dikarenakan, ketika mobil listrik menanjak semakin tinggi, maka *grade resistance* (R_g) yang melawan laju kendaraan menjadi lebih besar, sehingga kinerja motor listrik membutuhkan torsi besar.

Respon kecepatan saat pengujian tanjakan sistem kontrol tanpa FLC kecepatan rata-rata sebesar 174,90 RPM, sedangkan sistem kontrol FLC sebesar 195,00 RPM dengan *set-point* sebesar 200 RPM. Hal ini dikarenakan, ketika mobil listrik menanjak semakin tinggi, maka *grade resistance* (R_g) yang melawan laju kendaraan menjadi lebih besar. Akan tetapi, sistem kontrol tanpa FLC tidak ada umpan balik PWM yang mengakibatkan kecepatan melambat, sedangkan sistem kontrol FLC terdapat umpan balik penambahan PWM yang mengakibatkan kecepatan bertambah.

Respon kecepatan saat pengujian turunan sistem kontrol tanpa FLC kecepatan rata-rata sebesar 280,72 RPM, sedangkan sistem kontrol FLC sebesar 223,59 RPM dengan *set-point* sebesar 200 RPM. Hal ini dikarenakan, ketika mobil listrik menurun diuntungkan oleh *grade resistance* (R_g) yang membantu laju kendaraan. Akan tetapi, sistem kontrol tanpa FLC tidak ada umpan balik PWM yang mengakibatkan kecepatan bertambah, sedangkan sistem kontrol FLC terdapat umpan balik pengurangan PWM yang mengakibatkan kecepatan berkurang.

Penggunaan energi pengujian sistem kontrol tanpa FLC sebesar 38,464 km/kwh, sedangkan sistem kontrol FLC sebesar 44,39 km/kwh. Sehingga penggunaan energi sistem kontrol FLC lebih hemat dengan jarak tempuh yang lebih jauh.

SUMMARY

Speed Control and Monitoring System for Electric Cars Based on *Fuzzy Logic and Internet of Things*; Hendrik Iswanto; 161910201101; 2020; 121 pages; Department of Electrical Engineering, Faculty of Engineering, University of Jember.

New technological developments need to be developed to save fossil energy and to apply new energy sources, one of which is in the transportation sector which continues to develop, namely environmentally-friendly vehicles. This is triggered by increasing global greenhouse gas emissions in line with population growth and consumption of fossil fuels. Electric vehicles are one of the solutions in solving problems faced by the automotive sector around the world for environmental and energy sustainability. Industrial development is a challenge for engineers to be directly involved in designing tools that can be implemented in the industrial sector. Internet of Things (IoT) is an innovation that can collect and send sensor data via the internet which is stored in the cloud.

The use of the Internet of Things (IoT) technology in the TITEN EV-3 electric car monitoring system can be done via Blynk anywhere online and in real-time. The Blynk application is used for monitoring voltage, current, power, energy, angle, gas pedal, PWM, speed, and mileage of electric cars. The FLC system is used to manage the performance of the BLDC motor control when on uphill and downhill with the Mamdani type. The input variable is the potentiometer ADC value and the angle of the MPU 6050 sensor, while the output variable is PWM from the Arduino DUE. The speed of the electric car is controlled automatically by adding or decreasing the BLDC PWM control using the rule-base *fuzzy* logic controller algorithm.

The addition or subtraction of the PWM is controlled by the FLC system based on the input variable POT and angle. In a study with a road slope angle of 18.61 °, it is known that the grade of vehicle resistance when going up and down is

625.16 N. This is because, when an electric car goes up, the higher the resistance grade (R_g) against the vehicle's speed becomes greater so that the performance of the electric motor requires large torque.

The speed response when testing the incline of the control system without FLC has an average speed of 174.90 RPM, while the FLC control system is 195.00 RPM with a set-point of 200 RPM. This is because, when the electric car goes up, the higher the resistance grade (R_g) against the vehicle's speed is greater. However, the control system without FLC has no PWM feedback which causes the speed to slow down, while the FLC control system has additional PWM feedback which results in increased speed.

The speed response when testing the derivative of the control system without FLC, the average speed was 280.72 RPM, while the FLC control system was 223.59 RPM with a set-point of 200 RPM. This is because, when the electric car decreases, it benefits from the grade resistance (R_g) which helps the vehicle speed. However, the control system without FLC has no PWM feedback which results in increased speed, while the FLC control system has PWM reduction feedback which results in reduced speed.

The energy use of the control system testing without FLC was 38.464 km / kwh, while the FLC control system was 44.39 km / kwh. So that the energy use of the FLC control system is more efficient with a longer distance.

PRAKATA

Puji syukur ke hadirat Allah SWT atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi berjudul “Sistem Kontrol Kecepatan dan *Monitoring* Mobil Listrik Berbasis Logika *Fuzzy* dan *Internet of Things*”. Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan strata satu (S1) pada Jurusan Teknik Elektro Fakultas Teknik Universitas Jember. Penyusunan skripsi ini tidak lepas dari bantuan beberapa pihak. Oleh karena itu, penulis menyampaikan terimakasih kepada:

1. Bapak Dr. Triwahju Hardianto, ST., M.T. selaku Dekan Fakultas Teknik Universitas Jember;
2. Bapak Dr. Bambang Sri Kaloko, S.T.,M.T. selaku Ketua Jurusan Teknik Elektro Universitas Jember;
3. Bapak Prof. Dr. Ir. Bambang Sujanarko, M.M. selaku Dosen Pembimbing Utama serta Bapak Dr. Triwahju Hardianto, ST., M.T. selaku Dosen Pembimbing Anggota;
4. Bapak Ir. Widyono Hadi, M.T. selaku Dosen Penguji 1 serta Bapak Widya Cahyadi, S.T., M.T. selaku Dosen Penguji 2;
5. Bapak Alfredo Bayu Satriya S.T., M.T. selaku Dosen Pembimbing Akademik;
6. Bapak Sumarno dan Ibu Ida, selaku kedua orang tua penulis;
7. Semua pihak yang tidak dapat disebutkan satu per satu;

Penulis menyadari bahwa skripsi ini jauh dari kesempurnaan, untuk itu penulis menerima segala kritik dan saran dari semua pihak demi kesempurnaan skripsi ini, Terima Kasih.

Jember, 10 November 2020

Penulis

DAFTAR ISI

	Halaman
HALAMAN SAMPUL.....	i
HALAMAN JUDUL	ii
HALAMAN PERSEMBAHAN	iii
HALAMAN MOTTO	v
HALAMAN PERNYATAAN.....	vi
HALAMAN PEMBIMBING	vii
HALAMAN PENGESAHAN.....	viii
RINGKASAN/SUMMARY	ix
PRAKATA	xiii
DAFTAR ISI.....	xiv
DAFTAR TABEL	xviii
DAFTAR GAMBAR.....	xix
DAFTAR LAMPIRAN	xxii
BAB 1. PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	4
1.4 Tujuan Penelitian.....	4
1.5 Manfaat.....	4
BAB 2. TINJAUAN PUSTAKA.....	5
2.1 Mobil Listrik	5
2.2 <i>Internet of Things</i> (IoT)	6
2.3 Dinamika Kendaraan di Tanjakan dan Turunan	7
2.4 Motor <i>Brushless Direct Current</i>	8
2.4.1 Gambaran Umum Motor BLDC	8
2.4.2 Prinsip Kerja Motor BLDC.....	10
2.4.3 Sistem Komutasi Elektris Motor BLDC	13
2.5 Kontrol Kecepatan Motor BLDC.....	14

2.5.1 Inverter 3 Fasa	14
2.5.2 Metode <i>Six-step</i>	15
2.5.3 Dasar Pembangkitan sinyal PWM	16
2.5.4 Pengaturan Kecepatan dengan PWM	17
2.6 Fuzzy logic controller	18
2.6.1 Himpunan <i>Fuzzy</i>	18
2.6.2 Fungsi Keanggotaan	18
2.6.3 Operator Logika <i>Fuzzy</i>	20
2.6.4 <i>Fuzzy Inference System</i>	21
2.7 Arduino DUE.....	24
2.8 NodeMCU ESP8266.....	25
2.9 Sensor Sudut.....	26
2.10 Sensor Tegangan	27
2.11 Sensor Arus	28
2.12 Sensor Kecepatan.....	28
2.13 Baterai <i>Lithium Ion</i>.....	29
2.14 Software Arduino IDE.....	31
2.15 Aplikasi Blynk	32
BAB 3. METODE PENELITIAN.....	33
3.1 Tempat dan Waktu Penelitian.....	33
3.1.1 Tempat Penelitian	33
3.1.2 Waktu Penelitian.....	33
3.2 Alat dan Bahan Penelitian	33
3.3 Tahap Penelitian	34
3.4 Perancangan Alat.....	36
3.4.1 Perancangan Blok Diagram Sistem	36
3.4.2 Perancangan Skematik Sistem	37
3.4.3 Perancangan <i>Wiring</i> Diagram Sistem	39
3.4.4 Perancangan Sistem FLC	40
3.4.5 <i>Flowchart</i> Keseluruhan Sistem.....	53
3.5 Spesifikasi Motor BLDC	54
3.6 Spesifikasi Mobil Listrik	55
3.7 Perhitungan <i>Grade Resistance</i> Kendaraan.....	56
3.8 Desain GUI Blynk	58

3.9 Kalibrasi Sensor	59
3.9.1 Sensor Tegangan.....	59
3.9.2 Sensor Arus.....	60
3.9.3 Sensor Kecepatan	61
3.9.4 Sensor Sudut.....	62
3.9.5 Jarak Tempuh	63
3.10 Pengujian Alat.....	64
BAB 4. HASIL DAN PEMBAHASAN	65
4.1 Analisa Pengujian Sensor.....	65
4.1.1 Sensor Tegangan.....	65
4.1.2 Sensor Arus.....	66
4.1.3 Sensor Kecepatan.....	68
4.1.4 Sensor Sudut	69
4.1.5 Sensor Jarak Tempuh.....	71
4.2 Analisa Pengujian PWM.....	72
4.3 Analisa Pengujian GUI Blynk	74
4.3.1 Pengujian GUI Jarak Kirim 0-1 Kilometer	74
4.3.2 Pengujian GUI Jarak Kirim 1-2 Kilometer	75
4.4 Analisa Pengujian Sistem <i>Monitoring</i>	76
4.4.1 Pengujian Sistem <i>Monitoring</i> Jalan Lurus.....	76
4.4.2 Pengujian Sistem <i>Monitoring</i> Jalan Belok.....	79
4.4.3 Pengujian Sistem <i>Monitoring</i> Jalan Tanjakan	81
4.4.4 Pengujian Sistem <i>Monitoring</i> Jalan Turunan.....	83
4.5 Analisa Pengujian Respon Sistem Kontrol.....	86
4.5.1 Pengujian Respon Sistem Kontrol Tanjakan tanpa FLC	86
4.5.2 Pengujian Respon Sistem Kontrol Tanjakan FLC	87
4.5.3 Pengujian Respon Sistem Kontrol Turunan tanpa FLC.....	88
4.5.4 Pengujian Respon Sistem Kontrol Turunan FLC	89
4.6 Analisa Pengujian Penggunaan Energi Sistem Kontrol.....	90
4.6.1 Pengujian Penggunaan Energi Sistem Kontrol tanpa FLC	90
4.6.2 Pengujian Penggunaan Energi Sistem Kontrol FLC.....	92
BAB 5. KESIMPULAN DAN SARAN	96
5.1 Kesimpulan.....	96
5.2 Saran	97

DAFTAR PUSTAKA	98
LAMPIRAN.....	102



DAFTAR TABEL

	Halaman
Tabel 2.1 Spesifikasi Arduino DUE.....	25
Tabel 2.2 Spesifikasi modul NodeMCU ESP8266	26
Tabel 2.3 Kelebihan dan kekurangan baterai <i>lithium ion</i>	30
Tabel 3.1 Jadwal waktu penelitian	33
Tabel 3.2 Himpunan Parameter POT	42
Tabel 3.3 Himpunan parameter sudut	44
Tabel 3.4 Himpunan parameter PWM	46
Tabel 3.5 <i>Rule-base</i> sistem <i>fuzzy</i>	48
Tabel 3.6 Spesifikasi motor BLDC.....	55
Tabel 4.1 Pengujian Sensor Tegangan	65
Tabel 4.2 Pengujian sensor arus	67
Tabel 4.3 Pengujian sensor kecepatan	68
Tabel 4.4 Pengujian sensor sudut.....	70
Tabel 4.5 Pengujian sensor jarak tempuh	71
Tabel 4.6 Pengujian sinyal PWM.....	72
Tabel 4.7 Pengujian sistem <i>monitoring</i> jarak kirim 0-1 kilometer	75
Tabel 4.8 Pengujian sistem <i>monitoring</i> jarak kirim 1-2 kilometer	76
Tabel 4.9 Pengujian sistem <i>monitoring</i> jalan lurus.....	77
Tabel 4.10 Pengujian sistem <i>monitoring</i> jalan belok	79
Tabel 4.11 Pengujian sistem <i>monitoring</i> jalan tanjakan	81
Tabel 4.12 Pengujian sistem <i>monitoring</i> jalan turunan.....	84

DAFTAR GAMBAR

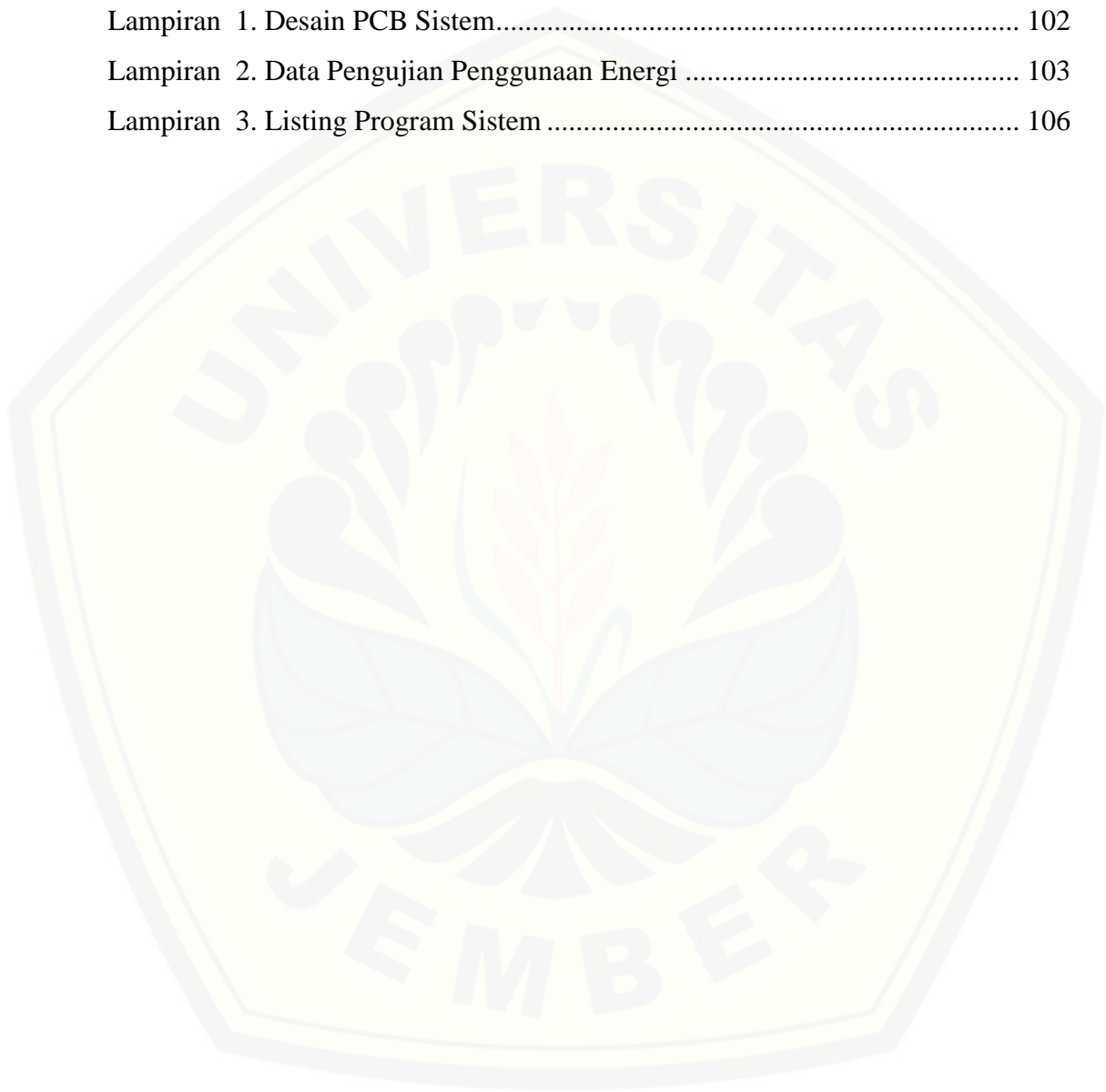
	Halaman
Gambar 2.1 Skema mobil listrik	5
Gambar 2.2 Arsitektur elemen IoT	6
Gambar 2.3 Arah <i>grade resistance</i> kendaraan	7
Gambar 2.4 Dinamika Kendaraan di tanjakan	8
Gambar 2.5 Konstruksi motor <i>Brushless Direct Current</i>	9
Gambar 2.6 Tegangan stator BLDC.....	12
Gambar 2.7 Sensor <i>hall effect</i> pada Motor BLDC.....	13
Gambar 2.8 Sinyal keluaran sensor <i>hall effect</i>	14
Gambar 2.9 Inverter 3 fasa.....	15
Gambar 2.10 Sinyal PWM dengan variasi <i>duty cycle</i>	16
Gambar 2.11 Metode penambahan PWM.....	17
Gambar 2.12 Representasi kurva linier.....	19
Gambar 2.13 Representasi kurva segitiga.....	19
Gambar 2.14 Representasi kurva trapesium.....	20
Gambar 2.15 Blok diagram sistem inferensi <i>fuzzy</i>	22
Gambar 2.16 Modul Arduino DUE.....	25
Gambar 2.17 Modul NodeMCU ESP8266	26
Gambar 2.18 Modul sensor GY-521 MPU 6050	27
Gambar 2.19 Resistor sebagai pembagi tegangan.....	27
Gambar 2.20 Modul sensor ACS712	28
Gambar 2.21 Mekanisme penempatan sensor <i>hall effect</i>	29
Gambar 2.22 Elemen baterai <i>lithium ion</i>	30
Gambar 2.23 Tampilan Arduino IDE.....	31
Gambar 2.24 Arsitektur <i>cloud</i> Blynk	32
Gambar 3.1 <i>Flowchart</i> penelitian	34
Gambar 3.2 Blok diagram sistem.....	36
Gambar 3.3 Skematik sistem.....	37

Gambar 3.4 Wiring diagram sistem	39
Gambar 3.5 Diagram sistem kontrol <i>fuzzy close-loop</i>	40
Gambar 3.6 <i>Fuzzy logic designer</i> pada Matlab	41
Gambar 3.7 <i>Membership function</i> variabel POT	42
Gambar 3.8 <i>Membership function</i> variabel sudut	44
Gambar 3.9 <i>Membership function</i> variabel PWM	46
Gambar 3.10 <i>Rule editor</i> pada Matlab	48
Gambar 3.11 <i>Rule viewer</i> pada Matlab	49
Gambar 3.12 <i>Membership function</i> variabel POT 200	50
Gambar 3.13 <i>Membership function</i> variabel sudut 15°	50
Gambar 3.14 <i>Membership function</i> variabel PWM 2,5	51
Gambar 3.15 <i>Surface viewer</i> pada matlab.....	52
Gambar 3.16 Flowchart keseluruhan sistem	53
Gambar 3.17 Desain mobil listrik TITEN EV-3	56
Gambar 3.18 <i>Grade resistance</i> kendaraan pada kemiringan 18,61°	57
Gambar 3.19 Desain GUI Blynk	58
Gambar 3.20 Grafik kalibrasi sensor tegangan	60
Gambar 3.21 Grafik kalibrasi sensor arus	61
Gambar 3.22 Grafik kalibrasi sensor kecepatan	62
Gambar 3.23 Grafik kalibrasi sensor sudut	63
Gambar 4.1 Grafik tegangan dan arus pengujian sistem <i>monitoring</i> jalan lurus..	78
Gambar 4.2 Grafik daya pengujian sistem <i>monitoring</i> jalan lurus	78
Gambar 4.3 Grafik tegangan dan arus pengujian sistem <i>monitoring</i> jalan belok .	80
Gambar 4.4 Grafik daya pengujian sistem <i>monitoring</i> jalan belok	81
Gambar 4.5 Grafik tegangan dan arus pengujian sistem <i>monitoring</i> jalan tanjakan	82
Gambar 4. 6 Grafik daya pengujian sistem <i>monitoring</i> jalan tanjakan.....	83
Gambar 4.7 Grafik tegangan dan arus pengujian sistem <i>monitoring</i> jalan turunan	85
Gambar 4.8 Grafik daya pada pengujian sistem <i>monitoring</i> jalan turunan.....	85
Gambar 4.9 Grafik respon kecepatan sistem kontrol tanjakan tanpa FLC	86

Gambar 4.10 Grafik respon kecepatan sistem kontrol tanjakan FLC	87
Gambar 4. 11 Grafik respon kecepatan sistem kontrol turunan tanpa FLC.....	88
Gambar 4.12 Grafik respon kecepatan sistem kontrol turunan FLC	89
Gambar 4.13 Grafik respon kecepatan pengujian penggunaan energi sistem kontrol tanpa FLC.....	90
Gambar 4.14 Grafik daya pengujian penggunaan energi sistem kontrol tanpa FLC	91
Gambar 4.15 Grafik penggunaan energi sistem kontrol tanpa FLC	92
Gambar 4.16 Grafik respon kecepatan pengujian penggunaan energi sistem kontrol FLC	93
Gambar 4.17 Grafik daya pengujian penggunaan energi sistem kontrol FLC.....	94
Gambar 4.18 Grafik penggunaan energi sistem kontrol FLC	95

DAFTAR LAMPIRAN

	Halaman
Lampiran 1. Desain PCB Sistem.....	102
Lampiran 2. Data Pengujian Penggunaan Energi	103
Lampiran 3. Listing Program Sistem	106



BAB 1. PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi baru perlu dikembangkan untuk melakukan penghematan energi fosil maupun untuk penerapan sumber energi baru, salah satunya di bidang transportasi yang terus berkembang yaitu kendaraan ramah lingkungan. Hal ini dipicu oleh meningkatnya emisi gas rumah kaca dunia seiring pertumbuhan populasi dan konsumsi bahan bakar fosil. Pada tahun 2016 emisi gas karbon dioksida (CO₂) sebesar 32,31 Gt CO₂ yang berasal dari bahan bakar fosil, 25 % dihasilkan dari sektor transportasi (Baatarbileg dan Lee, 2019). Dalam kaitan mencari solusi permasalahan tersebut, diadakan kompetisi tingkat perguruan tinggi untuk merancang dan membuat kendaraan hemat energi jenis *prototype* dan *urban concept*.

Shell Eco Marathon (SEM) merupakan kompetisi merancang dan membuat kendaraan hemat energi tingkat internasional dengan kategori mesin pembakaran dalam (MPD), baterai listrik, dan hidrogen (SEM,2019). Di Indonesia melalui Direktorat Kemahasiswaan, Direktorat Jenderal Pembelajaran dan Kemahasiswaan, Kementerian Riset Teknologi Dan Pendidikan Tinggi Republik Indonesia turut berperan yakni dengan menggali inovasi-inovasi di bidang penghematan energi di kendaraan dengan menyelenggarakan Kontes Mobil Hemat Energi (KMHE) dengan kategori mesin pembakaran dalam (MPD) dan baterai listrik (KMHE,2019). Karena itu, negara-negara maju seperti Amerika Serikat, China, Jepang, Korea dan beberapa negara di eropa menetapkan rencana meningkatkan penggunaan kendaraan listrik untuk mengurangi penggunaan bahan bakar fosil dibidang transportasi (Baatarbileg dan Lee, 2019).

Kendaraan listrik menjadi salah satu solusi dalam pemecahan masalah yang dihadapi oleh sektor otomotif di seluruh dunia untuk kelestarian lingkungan dan energi (Zhang dan Li, 2020). Dalam satu dekade terakhir, kendaraan listrik sebagai transportasi alternatif pengganti kendaraan mesin pembakaran dalam (MPD) (Vdovic dkk., 2019). Menurut badan energi internasional (IEA), pada tahun 2018 penjualan mobil listrik mencapai 5,1 juta unit atau meningkat 63% dari tahun 2017

(Till Bunsen dkk., 2019). Menurut Menteri Perindustrian (2019), pada tahun 2030 menargetkan Indonesia menjadi pusat kendaraan listrik ASEAN.

Perkembangan industri menjadi tantangan bagi para insinyur untuk terlibat langsung dalam merancang perangkat yang bisa diimplementasikan di sektor industri. Revolusi industri 4.0 menjadi solusi tantangan tersebut, bahwa *Internet of Things* (IoT) dan kecerdasan buatan (AI) suatu prioritas (Giay dan Adiono, 2019). Menurut Kushwah and Rani (2019) *Internet of Things* (IoT) adalah suatu sistem yang dapat berinteraksi satu dengan yang lain, pengguna dengan komputer, dan mesin dengan mesin melalui internet. *Internet of Things* (IoT) merupakan sebuah inovasi yang dapat mengumpulkan dan mengirim data sensor melalui internet yang disimpan di *cloud* (Padmaja dkk., 2019).

Gerak kendaraan saat melaju dihasilkan oleh gaya dorong roda depan (F_f) dan gaya dorong roda belakang (F_r). Kendaraan yang sedang melaju terdapat 3 gaya hambat, yaitu *Aerodynamic Resistance* (R_a), *Rolling Resistance* (R_r), dan *Grade Resistance* (R_g). Saat kendaraan melaju pada lintasan tanjakan akan meningkatkan torsi mesin, tetapi kecepatan kendaraan berkurang dikarenakan adanya *grade resistance* yang melawan arah laju kendaraan, sedangkan saat di lintasan turunan diuntungkan karena adanya bantuan *grade resistance* (User dan Tabanli, 2017). Konsumsi bahan bakar kendaraan saat di tanjakan meningkat, sedangkan saat di turunan berkurang. Hasil penelitian dengan simulasi konsumsi kendaraan saat di tanjakan sebesar 878,4 mg, sedangkan saat di turunan sebesar 834,2 mg (Bhoi dkk., 2013).

Penelitian *monitoring* kendaraan listrik berbasis *Internet of Things* (IoT) yang dikembangkan di *Department of Electrical and Electronics Engineering, Pamukkale University, Turkey* dengan sistem *monitoring* menggunakan modul NRF24L01 untuk mengirimkan data ke pengguna yang ditampilkan *Thin Film Transistor Liquid Crystal Display*. Akan tetapi dalam penelitian tersebut memiliki keterbatasan jarak pengiriman data yaitu 1000 meter (Kilickir dkk., 2017).

Penelitian sistem *monitoring* dan kontrol pada kendaraan listrik di Indonesia khususnya di Universitas Jember yaitu dengan berjudul “Sistem *Monitoring* Mobil Listrik Universitas Jember Berpenggerak Motor BLDC Berbasis Radio Frekuensi”.

Penelitian tersebut menggunakan modul *Radio Telemetry* untuk mengirim data ke pengguna yang ditampilkan di komputer menggunakan *Software Visual Studio*. Akan tetapi dalam penelitian tersebut memiliki keterbatasan jarak pengiriman data yaitu 300 meter (Sujanarko dkk., 2019).

Pengembangan mobil listrik oleh tim mobil listrik Universitas Jember (TITEN Electric Vehicle) memiliki kendala dalam pengukuran tegangan, arus, daya, energi, sudut, pedal gas, PWM, kecepatan, dan jarak tempuh. Sistem *monitoring* dilakukan secara manual dan berhubungan langsung dengan objek sehingga kurang efektif. Mengacu pada penelitian sebelumnya, penulis ingin mengembangkan penelitian yang berjudul “Sistem Kontrol Kecepatan dan *Monitoring* Mobil Listrik Berbasis Logika *Fuzzy* dan *Internet of Things*”.

Dalam prosesnya, digunakan kontrol kecepatan mobil listrik saat di tanjakan dan turunan dengan metode *fuzzy logic controller* dan sistem *monitoring* dengan media komunikasi *wireless fidelity* (Wi-Fi) yaitu modul NodeMCU ESP8266 yang dapat dipantau melalui aplikasi Blynk secara *real time*.

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang diatas, dapat dirumuskan beberapa rumusan masalah diantaranya :

1. Bagaimana perancangan sistem kontrol kecepatan dan *monitoring* mobil listrik TITEN EV-3 ?
2. Bagaimana kinerja perangkat sistem *monitoring* mobil listrik TITEN EV-3 berbasis *Internet of Things* ?
3. Bagaimana respon kecepatan saat pengujian tanjakan sistem kontrol FLC dan tanpa FLC ?
4. Bagaimana respon kecepatan saat pengujian turunan sistem kontrol FLC dan tanpa FLC ?
5. Bagaimana penggunaan energi saat pengujian sistem kontrol FLC dan tanpa FLC ?

1.3 Batasan Masalah

Berdasarkan uraian rumusan masalah diatas, maka ada beberapa batasan masalah dalam skripsi ini diantaranya :

1. Motor BLDC yang digunakan berdaya 250 Watt dengan tegangan 36 Volt dan tidak membahas secara detail konstruksi motor BLDC.
2. *Monitoring* melalui aplikasi Blynk dengan Arduino DUE untuk memproses data.
3. Pengujian dinamis dilakukan di lingkungan Universitas Jember.
4. Pengujian tanjakan dan turunan dengan sudut terbesar sebesar $18,61^\circ$.
5. Massa mobil listrik dan joki tidak melebihi 200 Kg.

1.4 Tujuan Penelitian

Berdasarkan uraian rumusan masalah diatas, maka tujuan dari penelitian ini adalah sebagai berikut :

1. Mendesain sistem kontrol kecepatan dan *monitoring* mobil listrik TITEN EV-3.
2. Mengetahui kinerja perangkat sistem *monitoring* mobil listrik TITEN EV-3 berbasis *Internet of Things*.
3. Mengetahui respon kecepatan saat pengujian tanjakan sistem kontrol FLC dan tanpa FLC.
4. Mengetahui respon kecepatan saat pengujian turunan sistem kontrol FLC dan tanpa FLC.
5. Mengetahui penggunaan energi saat pengujian sistem kontrol FLC dan tanpa FLC.

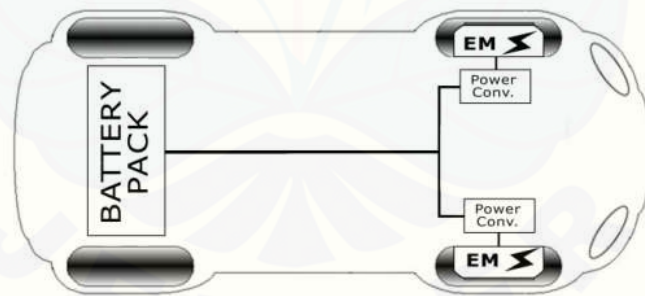
1.5 Manfaat

Adapun manfaat dari penelitian yang diharapkan setelah tujuan ini tercapai adalah dapat menerapkan kontrol kecepatan otomatis saat di tanjakan maupun turunan dan menerapkan revolusi industri 4.0 sebagai pengembangan teknologi mobil listrik serta sarana pemicu penggunaan mobil listrik di Indonesia.

BAB 2. TINJAUAN PUSTAKA

2.1 Mobil Listrik

Mobil listrik adalah mobil yang menggunakan motor listrik sebagai penggerak utama dengan energi listrik yang disimpan dalam baterai maupun penyimpanan energi lainnya. Mobil listrik pertama kali dibuat oleh Thomas Davenport dan Robert Anderson pada tahun 1830-an. Namun pada saat itu menggunakan baterai yang belum bisa diisi ulang. Pada tahun 1881 Gustave Trouve mengembangkan baterai yang dapat diisi ulang. Akan tetapi, pada tahun 1918 Henry Ford berhasil membuat mobil pembakaran dalam. Dibandingkan dengan mobil listrik, mobil pembakaran dalam memiliki harga yang lebih murah, pengisian bahan bakar yang mudah dan memiliki jarak tempuh yang lebih panjang. Akibatnya mobil listrik mengalami penurunan permintaan pada tahun 1930-an. Seiring berjalannya waktu muncul permasalahan polusi udara dan menipisnya energi bahan bakar fosil. Pada saat terjadinya krisis energi tahun 1970-an memunculkan kembali minat mobil listrik (Hosseinpour dkk., 2015).



Gambar 2.1 Skema mobil listrik

(Sumber : Mahmoudi, Flah, and Sbita 2014)

Mobil listrik memiliki banyak keunggulan dibandingkan dengan mobil pembakaran dalam konvensional. Seperti tidak menghasilkan emisi kendaraan bermotor, efisiensi tinggi, dan tidak tergantung bahan bakar fosil dan pengoperasiannya mudah dan halus. Mobil listrik memiliki tiga komponen utama yaitu motor listrik, pengendali kecepatan, dan sistem penyimpanan energi. Motor

listrik digunakan sebagai penggerak utama kendaraan. Pengendalian kecepatan motor digunakan untuk mengendalikan kecepatan motor. Sistem penyimpanan energi adalah komponen penting dalam mobil listrik. Pemilihan sistem penyimpanan energi listrik harus memperhatikan beberapa fitur vital seperti kapasitas, siklus hidup, dan kepadatan energi (Wong, 2017).

Dalam satu dekade terakhir, kendaraan listrik sebagai transportasi alternatif pengganti kendaraan mesin pembakaran dalam (MPD) (Vdovic dkk., 2019). Menurut badan energi internasional (IEA), pada tahun 2018 penjualan mobil listrik mencapai 5,1 juta unit atau meningkat 63% dari tahun 2017 (Till Bunsen dkk., 2019). Menurut Menteri Perindustrian (2019), pada tahun 2030 menargetkan Indonesia menjadi pusat kendaraan listrik ASEAN.

2.2 Internet of Things (IoT)

Internet of Things (IoT) adalah suatu sistem yang dapat berinteraksi satu dengan yang lain, pengguna dengan komputer, dan mesin dengan mesin melalui internet (Kushwah dan Rani, 2019). Konsep IoT cukup sederhana dengan tiga elemen penting yaitu *hardware* yang dilengkapi dengan modul IoT, perangkat koneksi ke internet seperti modem atau *router wireless* dan *cloud data server* yang digunakan untuk tempat penyimpanan aplikasi dan *database*.



Gambar 2.2 Arsitektur elemen IoT

(Sumber : Setiawan, Tanudjaja, and Octaviani 2019)

Konsep IoT menjadikan fungsi internet mendalam dan meluas, dengan memungkinkan untuk berinteraksi dengan perangkat *hardware* seperti peralatan rumah tangga, sensor dan aktuator, kamera pengintai, kendaraan dan lainnya melalui internet.

2.3 Dinamika Kendaraan di Tanjakan dan Turunan

Gerak kendaraan saat melaju dihasilkan oleh gaya dorong roda depan (F_f) dan gaya dorong roda belakang (F_r). Kendaraan yang sedang melaju terdapat 3 gaya hambat, yaitu *Aerodynamic Resistance* (R_a), *Rolling Resistance* (R_r), dan *Grade Resistance* (R_g). *Aerodynamic resistance* adalah gaya hambat yang terjadi karena adanya hambatan udara. *Rolling resistance* adalah gaya hambat akibat gesekan ban dengan jalan. *Grade resistance* adalah gaya hambat yang membantu atau melawan laju kendaraan karena dipengaruhi oleh sudut kemiringan jalan.



Gambar 2.3 Arah *grade resistance* kendaraan

(Sumber : Bhoi dkk., 2013)

Saat kendaraan melaju pada lintasan tanjakan akan meningkatkan torsi mesin sehingga kecepatan kendaraan berkurang dikarenakan adanya *grade resistance* yang melawan arah laju kendaraan yang dinyatakan kemiringan positif, sedangkan saat di lintasan turunan diuntungkan adanya bantuan *grade resistance* yang dinyatakan kemiringan negatif. *Grade resistance* juga mempengaruhi *rolling resistance*, saat di tanjakan *rolling resistance* akan semakin besar dan sebaliknya jika saat di turunan *rolling resistance* semakin kecil.

Besar *grade resistance* dapat dihitung dengan rumus berikut (User dan Tabanli, 2017) :

$$R_g = \pm m \times g \times \sin \vartheta \quad (2.1)$$

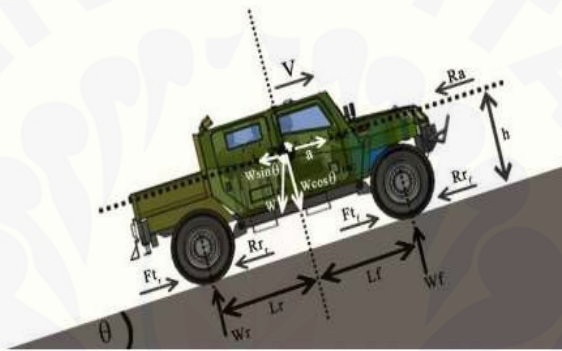
Keterangan :

R_g = *Grade resistance* (N)

m = Massa Kendaraan (Kg)

g = Percepatan Gravitasi (m/s^2)

ϑ = Sudut Miring ($^\circ$)



Gambar 2.4 Dinamika Kendaraan di tanjakan

(Sumber : Siregar dan Sutantra, 2018)

Konsumsi bahan bakar kendaraan meningkat saat di tanjakan, sedangkan berkurang saat di turunan. Hasil simulasi konsumsi kendaraan saat di tanjakan sebesar 878,4 mg, sedangkan saat di turunan sebesar 834,2 mg (Bhoi dkk., 2013).

2.4 Motor *Brushless Direct Current*

2.4.1 Gambaran Umum Motor BLDC

Motor listrik adalah alat yang digunakan untuk mengubah energi listrik menjadi energi mekanik. Motor BLDC dalam komutasinya tidak menggunakan sikat mekanis, tetapi dilakukan secara elektronik yang dikontrol. Sehingga dapat mengurangi percikan listrik dan mengurangi gangguan elektromagnetik atau *Electromagnetic Interference* (EMI). Motor BLDC disebut juga motor sinkron,

dikarenakan medan magnet stator dan medan magnet rotor berputar pada frekuensi yang sama. Sehingga pada motor BLDC tidak mengalami slip antara stator dan rotor. Interaksi medan magnet yang dihasilkan stator dengan permanen medan magnet rotor akan menyebabkan motor berputar. Agar motor BLDC terus berputar dengan mengubah arus stator dengan pola tertentu (Ikhlas, 2015). Motor BLDC memiliki beberapa keuntungan dibandingkan dengan motor *Direct Current* konvensional diantaranya :

- 1) Memiliki tingkat efisiensi yang tinggi
- 2) Memiliki torsi dan kecepatan tinggi
- 3) Emisi suara yang lebih rendah
- 4) Perawatan mudah



Gambar 2.5 Konstruksi motor *Brushless Direct Current*

(Sumber : Chaudhari et al. 2017)

Secara umum motor BLDC memiliki dua komponen utama yaitu stator dan rotor. Stator adalah bagian pada motor yang diam yang terdiri dari kumparan atau laminasi baja yang bertumpuk pada slot. Stator berfungsi sebagai medan putar motor secara elektromagnetik pada rotor. Rotor adalah bagian yang berputar akibat adanya gaya elektromagnetik dari stator dan terbuat dari magnet permanen (Ikhlas, 2015).

Motor BLDC menggunakan komutasi elektrik yaitu dengan menggunakan sensor *hall effect*. Sehingga dalam komutasinya harus memiliki ketepatan waktu yang sesuai agar dapat berputar terus-menerus (Cladella, 2017). Sensor *hall effect* digunakan untuk mendeteksi adanya medan magnet rotor, dan dipasang di stator yang akan mengirimkan sinyal untuk proses komutasi elektronik. Motor BLDC berputar dengan hanya dua fasa yang tersuplai sementara yang satu tidak tersuplai (Poovizhi dkk., 2017).

2.4.2 Prinsip Kerja Motor BLDC

Motor BLDC bekerja dengan sumber tegangan *Direct Current* yang kemudian diubah menjadi tegangan AC 3 fasa melalui inverter daya dengan sudut antar fasa sebesar 120° . Adanya interaksi magnet elektrik pada stator dengan magnet permanen rotor, sehingga arus yang melewati stator timbul medan magnet (B).

$$B = \frac{\mu N \iota}{2i} \quad (2. 2)$$

Keterangan :

N = Jumlah lilitan

B = Medan magnet (Wb/m^2)

ι = Panjang lilitan (m)

i = Arus (A)

μ = Permeabilitas bahan (Wb/Am)

Arus yang melewati stator memiliki nilai yang berubah setiap saat dengan beda fasa yang sama diantara ketiga fasa sehingga menyerupai sumber tiga fasa. Akibat perbedaan fasa dan perubahan arus, maka medan magnet setiap kumparan berubah secara teratur.

Sehingga timbul interaksi medan magnet stator dengan medan magnet permanen rotor dari perubahan polarisasi tersebut menyebabkan medan putar magnet (Cladella, 2017).

$$N_s = \frac{120 \times f}{p} \quad (2.3)$$

Keterangan :

N_s = Kecepatan (rpm)

f = Frekuensi (Hz)

p = Jumlah kutub dan 120 didapat dengan 1 putaran penuh dalam 3 fasa motor

Hubungan torsi dengan daya dapat dihitung dengan persamaan sebagai berikut (Reddy Avutu dkk., 2020).

$$P = \omega \times T \quad (2.4)$$

Keterangan :

P = Daya (W)

ω = Kecepatan sudut (rad/s)

T = Torsi (Nm)

Kecepatan sudut motor dapat dihitung dengan persamaan sebagai berikut :

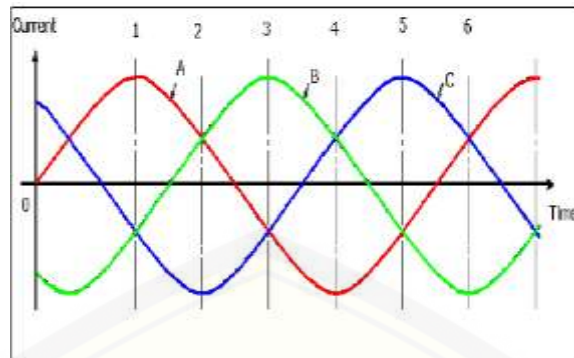
$$\omega = \frac{2\pi \times N_r}{60} \quad (2.5)$$

Keterangan :

ω = Kecepatan sudut (rad/s)

N_r = Kecepatan motor (RPM)

Perubahan polaritas tersebut ditunjukkan pada gambar 2.6 yaitu bentuk sinyal tegangan 3 fasa pada stator BLDC.



Gambar 2.6 Tegangan stator BLDC

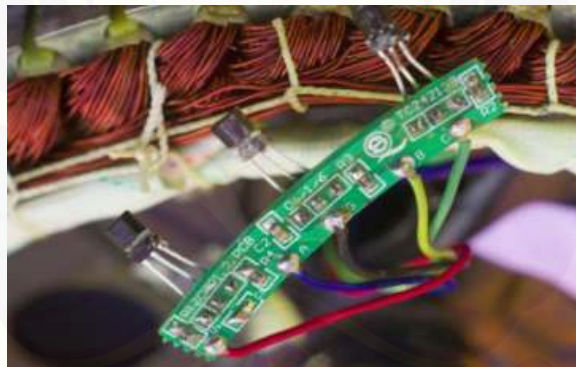
(Sumber : Ismail 2019)

Ketika stator A diberikan tegangan negatif, maka akan timbul medan magnet stator dengan polaritas negatif sedangkan stator B dan C yang diberikan tegangan positif akan memiliki polaritas positif. Adanya perbedaan polaritas medan magnet stator dengan medan magnet permanen rotor, maka sisi positif medan magnet permanen rotor akan berputar mendekati medan magnet stator A, sedangkan sisi negatif akan berputar mengikuti medan magnet stator B dan C. Tegangan yang digunakan berupa AC, maka medan magnet stator akan berubah-ubah polaritasnya dan besarnya sesuai perubahan tegangan AC. Ketika stator B dan C memiliki medan magnet negatif akibat mendapat tegangan negatif dan stator C memiliki medan magnet positif akibat mendapat positif, maka magnet permanen rotor akan berputar menuju polaritas yang berlawanan yaitu sisi negatif akan mendekati medan magnet stator C dan sisi positif akan berputar menuju medan magnet stator A dan B.

Ketika stator B memiliki medan magnet negatif akibat diberikan tegangan negatif dan Stator A dan C memiliki medan magnet positif akibat diberikan tegangan positif, maka sisi positif magnet permanen akan berputar menuju stator B dan sisi negatif akan menuju stator A dari stator C. Karena tegangan AC digunakan secara kontinyu, proses perubahan polaritas tegangan stator ini secara terus-menerus sehingga menciptakan medan magnet stator dan medan magnet permanen rotor akan berputar mengikuti medan putar magnet stator. Hal inilah yang menyebabkan rotor pada motor BLDC dapat berputar (Dharmawan, 2009).

2.4.3 Sistem Komutasi Elektris Motor BLDC

Motor BLDC tidak menggunakan sikat mekanis (*brush*) dalam sistem komutasinya, tetapi menggunakan sistem komutasi elektrik. Komutasi merupakan perubahan fasa arus motor pada waktu yang tepat untuk menghasilkan torsi rotasional. Sehingga diperlukan ketepatan waktu yang baik agar motor dapat berputar terus-menerus dengan kecepatan dan torsi yang konstan. Terdapat dua metode untuk menentukan waktu perubahan komutasi yaitu dengan sensor (*sensored*) dan tanpa sensor (*sensorless*). Metode tanpa sensor untuk komutasinya dengan mendeteksi aktivitas elektrik dari tiap belitan. Salah satunya dengan pendeteksian *zero crossing point* dengan menggunakan *back-EMF*. Metode sensor digunakan komponen tambahan yaitu sensor *hall effect* dan *optical encoder*. Dalam komutasinya dengan mendeteksi posisi rotor dengan memberikan sinyal tinggi atau rendah untuk merubah gerak komutasi (Wicaksono 2016;Cladella 2017).



Gambar 2.7 Sensor *hall effect* pada Motor BLDC

(Sumber : APRESCO 2017)

Secara umum untuk sistem komutasi elektrik motor BLDC dengan menggunakan metode sensor. Pada metode ini yang paling sering digunakan adalah sensor *hall effect*. Sensor *hall effect* digunakan untuk mendeteksi medan magnet permanen rotor untuk menentukan posisi rotor dan akan mengirimkan sinyal elektrik. Sensor *hall effect* ditempatkan setiap 120° pada jarak antar stator dengan tujuan agar deteksi terhadap vektor fluks stator akurat setiap perpindahan komutasinya.

Gambar 2.8 menunjukkan hasil sinyal keluaran dari sensor *hall effect*. Perbedaan sinyal keluaran antar sensor akan mempengaruhi proses *switching* inverter sehingga tegangan yang masuk ke motor BLDC akan berbeda-beda sesuai perubahan posisi rotor.



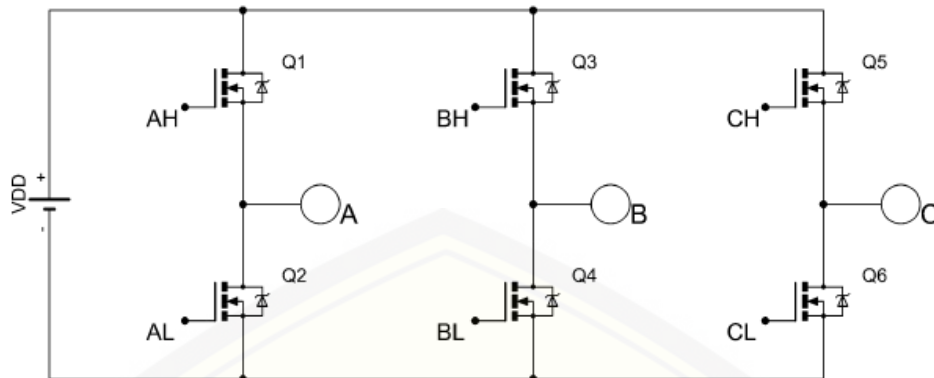
Gambar 2.8 Sinyal keluaran sensor *hall effect*

(Sumber : (Cladella, 2017))

2.5 Kontrol Kecepatan Motor BLDC

2.5.1 Inverter 3 Fasa

Inverter 3 fasa merupakan rangkaian elektronika daya yang berfungsi mengubah tegangan searah (DC) menjadi tegangan bolak-balik (AC). Inverter 3 fasa terdiri dari 6 komponen *switching*. Terdapat 2 jenis mode operasi yang biasa digunakan pada kontrol motor BLDC yaitu mode konduksi 120° dan mode konduksi 180° . Rangkaian inverter 3 fasa dengan 6 buah MOSFET n-channel ditunjukkan pada gambar 2.9.



Gambar 2.9 Inverter 3 fasa

(Sumber :Ismail, 2019)

Kumparan stator motor BLDC yang aktif setiap step-nya hanya satu buah kumparan “on” dan satu buah kumparan lainnya “off” atau tidak teraliri listrik. Hal ini menyebabkan tiap stepnya hanya diperbolehkan 2 buah MOSFET saja yang aktif secara bergantian pada mode konduksi 120° . Inverter 3 fasa mode konduksi 120° memungkinkan setiap komponen *switching* akan terkonduksi selama 120° dengan pasangan konduksi yang berbeda, 60° pertama antara AH-BL, dan 60° kedua antara AH-CL, kemudian BH-CL, BH-AL, CH-AL, CH-BL, AH-BL dan siklus berulang kembali.

2.5.2 Metode *Six-step*

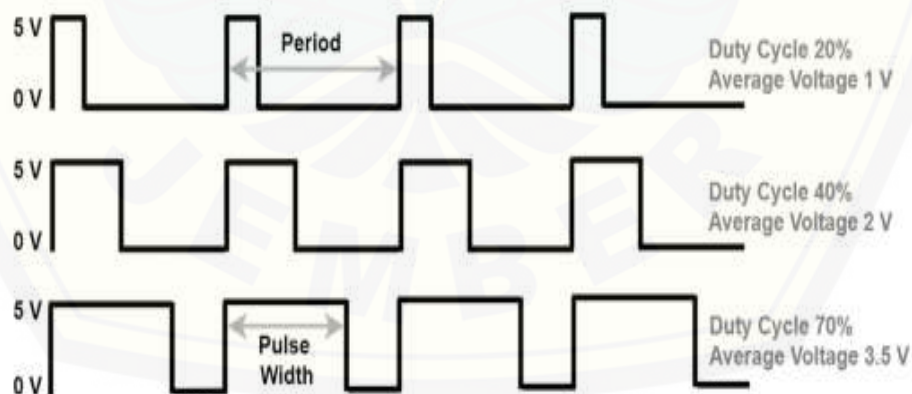
Metode *Six-Step* merupakan metode yang digunakan untuk pengendalian motor BLDC. Metode *six-step* sederhana sehingga mudah diimplementasikan. Akan tetapi, metode ini memiliki kelemahan yaitu memiliki suara yang bising, dikarenakan sinyal PWM yang digunakan dalam metode *six-step* merupakan sinyal PWM berbentuk kotak dengan frekuensi tertentu sehingga menciptakan gelombang AC yang berbentuk kotak. Akibat dari gelombang yang berbentuk kotak timbul gelombang harmonik. Oleh sebab itu gelombang harmonik inilah yang menyebabkan motor BLDC bising (Ismail, 2019).

2.5.3 Dasar Pembangkitan Sinyal PWM

Pulse Width Modulation (PWM) adalah suatu teknik modulasi yang mengubah lebar pulsa dalam suatu periode agar memperoleh tegangan rata-rata yang berbeda dengan amplitudo dan frekuensi tetap. Sinyal PWM dapat dibangkitkan dengan metode analog yaitu dengan rangkaian op-amp atau dengan metode digital yaitu menggunakan mikrokontroler. Adapun dengan metode analog perubahan PWM sangat halus, sedangkan dengan metode digital perubahan PWM dipengaruhi oleh resolusi PWM. Resolusi merupakan jumlah variasi perubahan nilai dalam PWM, misalnya suatu PWM memiliki resolusi 8 bit berarti PWM memiliki variasi perubahan nilai sebanyak $2^8 = 256$ variasi yaitu mulai dari 0-255 yang dapat dinyatakan dalam *duty cycle* 0-100% dari keluaran PWM tersebut (Masudi, 2014). Perhitungan *duty cycle* PWM dengan mengatur lebar pulsa “on” dan “off” dalam satu periode gelombang.

$$Duty\ cycle = \frac{T_{on}}{T_{on} + T_{off}} \times 100\ \% \quad (2.6)$$

Sinyal PWM dengan *duty cycle* yang berbeda-beda dapat dilihat pada gambar 2.10.

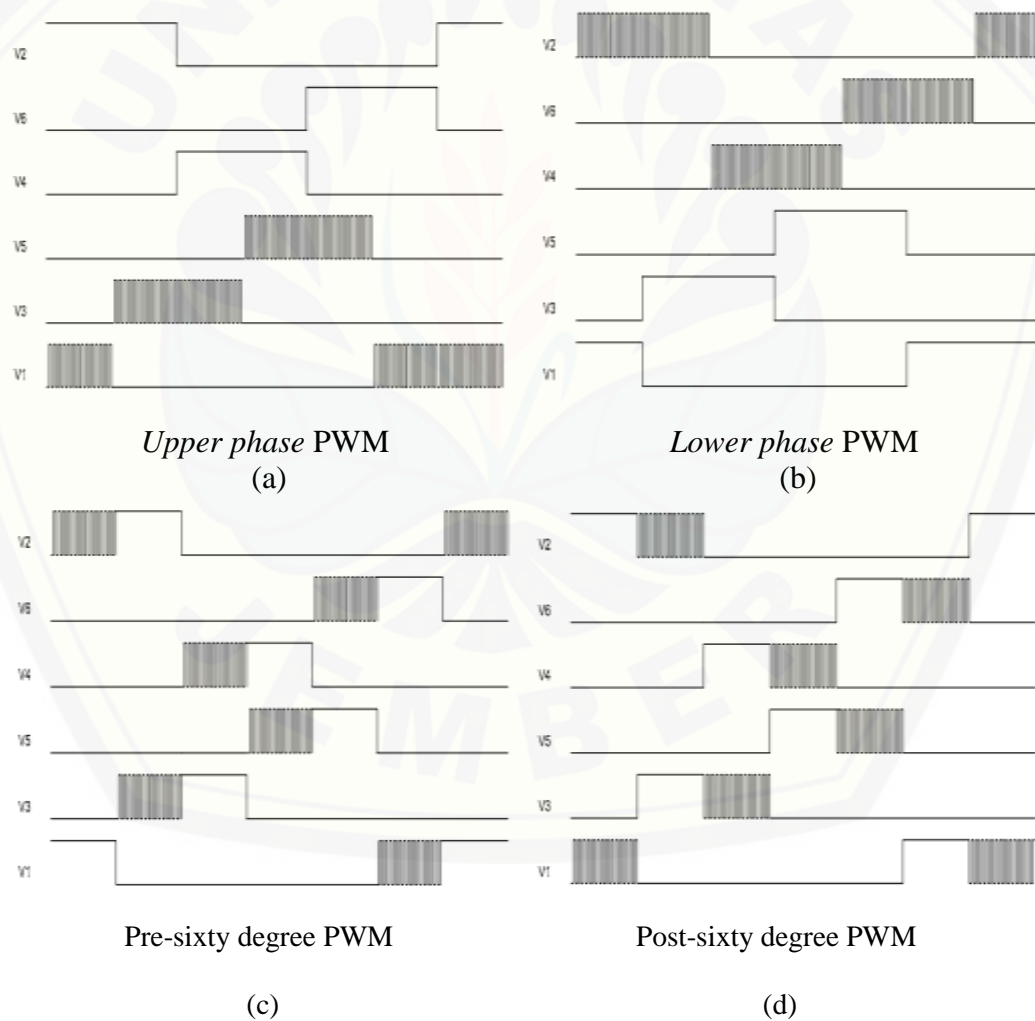


Gambar 2.10 Sinyal PWM dengan variasi duty cycle

(Sumber : Conker, 2019)

2.5.4 Pengaturan Kecepatan dengan PWM

Metode PWM merupakan metode yang cukup efektif untuk mengendalikan kecepatan motor. Cara kerjanya dengan mengatur persentase *duty cycle*, yaitu dengan mengatur persentase lebar pulsa *high* terhadap periode dari suatu sinyal persegi dalam bentuk tegangan periodik yang diberikan ke motor. Tegangan keluaran inverter dapat dikendalikan dengan mengubah *duty cycle*, sehingga bisa digunakan untuk mengontrol kecepatan motor BLDC. Terdapat empat metode untuk menambahkan PWM pada sistem *switching* inverter tiga fasa yaitu dapat dilihat gambar 2.11.



Gambar 2.11 Metode penambahan PWM

(Sumber :Ismail 2019)

2.6 Fuzzy Logic Controller

Fuzzy yang berarti samar, kabur atau tidak jelas yang dikembangkan pertama kali oleh Lotfi Zadeh pada tahun 1965. Dasar logika *fuzzy* adalah himpunan *fuzzy* artinya derajat keanggotaan dengan batasan yang tidak jelas atau samar sebagai penentu elemen dalam suatu himpunan yang memiliki nilai interval antara 0 dan 1 (Alazzam dan Ahmed, 2019). FLC dalam kerjanya didasarkan pada aturan linguistik yang menghubungkan *input* dan *output*. Motivasi terbesar untuk menggunakan sistem kendali *fuzzy* adalah untuk mendeskripsikan masalah dunia nyata yang rumit ke dalam sebuah model tetapi dalam bentuk linguistik tergantung pada aturan IF-THEN, semua itu berarti bahwa teknik pengendalian ini memiliki keuntungan yang sangat luar biasa yang akan mudah untuk memahami dan menerapkan seperti mengkategorikan kecepatan sebagai rendah, sedang dan cepat (Pooja Agarwal, 2013).

2.6.1 Himpunan *Fuzzy*

Jika X adalah koleksi dari obyek-obyek yang dinotasikan secara generik oleh x , maka suatu himpunan *fuzzy* \tilde{A} , dalam X adalah suatu himpunan pasangan berurutan :

$$\tilde{A} = \{(x, \mu_{\tilde{A}}(X)) \mid x \in X\} \quad (2.7)$$

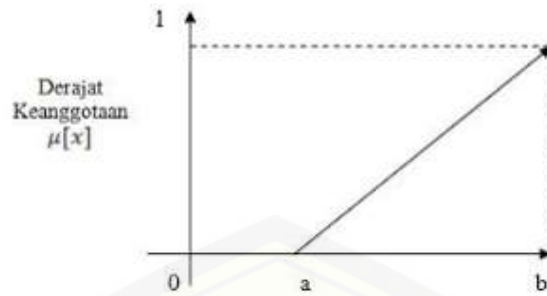
Dengan $\mu_{\tilde{A}}(X)$ adalah derajat keanggotaan x di \tilde{A} yang memetakan X ke ruang keanggotaan M yang terletak pada rentang $[0,1]$.

2.6.2 Fungsi Keanggotaan

Membership function atau *membership function* adalah suatu kurva yang menunjukkan pemetaan titik-titik *input* data ke dalam nilai keanggotaan adalah dengan melalui pendekatan fungsi. Ada beberapa fungsi yang bisa digunakan.

a. Representasi Linier

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



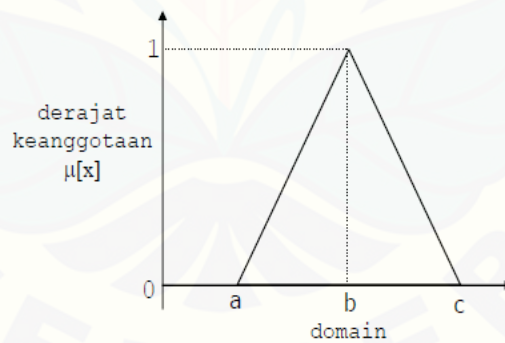
Gambar 2.12 Representasi kurva linier

Membership function:

$$\mu(X) = \begin{cases} 0 & ; x \leq a \\ (x-a)/(b-a); & a \leq x \leq b \\ 1 & ; x \geq b \end{cases} \quad (2.8)$$

b. Representasi Kurva Segitiga

Kurva segitiga pada dasarnya merupakan gabungan antara 2 garis (linear)



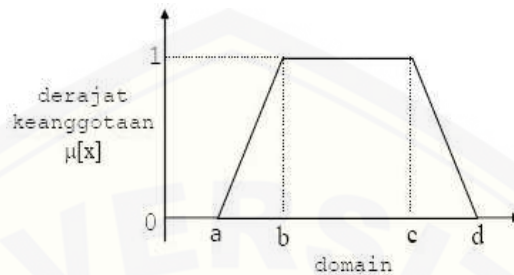
Gambar 2.13 Representasi kurva segitiga

Membership function:

$$\mu(X) = \begin{cases} 0 & ; x \leq a \text{ atau } x \geq c \\ (x-a)/(b-a); & a \leq x \leq b \\ (b-x)/(c-b); & b \leq x \leq c \end{cases} \quad (2.9)$$

c. Representasi Kurva Trapesium

Kurva trapesium pada dasarnya seperti bentuk segitiga, hanya saja ada beberapa titik yang memiliki nilai keanggotaan 1.



Gambar 2.14 Representasi kurva trapesium

Membership function:

$$\mu(X) = \begin{cases} 0 & ; x \leq a \text{ atau } x \geq d \\ (x-a)/(b-a) & ; a \leq x \leq b \\ 1 & ; b \leq x \leq c \\ (d-x)/(d-c) & ; x \geq d \end{cases} \quad (2.10)$$

2.6.3 Operator Logika Fuzzy

a. Operator Dasar

Seperti halnya himpunan konvensional, ada beberapa operasi yang didefinisikan secara khusus untuk mengkombinasi dan memodifikasi himpunan fuzzy. Nilai keanggotaan sebagai hasil dari operasi 2 himpunan sering dikenal dengan nama *fire strength* atau α -predikat. Ada tiga operator dasar yang diciptakan oleh Zadeh yaitu AND, OR, dan NOT.

b. Operator AND

Operator ini berhubungan dengan operasi interseksi atau irisan pada himpunan α -predikat sebagai hasil operasi dengan operator AND diperoleh dengan mengambil nilai keanggotaan terkecil antar elemen pada himpunan-himpunan yang bersangkutan.

$$\mu_{A \cap B} = \min(\mu_A(x), \mu_B(y)) \quad (2.11)$$

c. Operator OR

Operator ini berhubungan dengan operasi union atau gabungan pada himpunan. α -predikat sebagai hasil operasi dengan operator OR diperoleh dengan mengambil nilai keanggotaan terbesar antar elemen pada himpunan-himpunan yang bersangkutan.

$$\mu_{A \cup B} = \max(\mu_A(x), \mu_B(y)) \quad (2.12)$$

d. Operator NOT

Operator ini berhubungan dengan operasi komplemen pada himpunan. α -predikat sebagai hasil operasi dengan operasi NOT diperoleh dengan mengurangi nilai keanggotaan elemen pada himpunan yang bersangkutan dari 1.

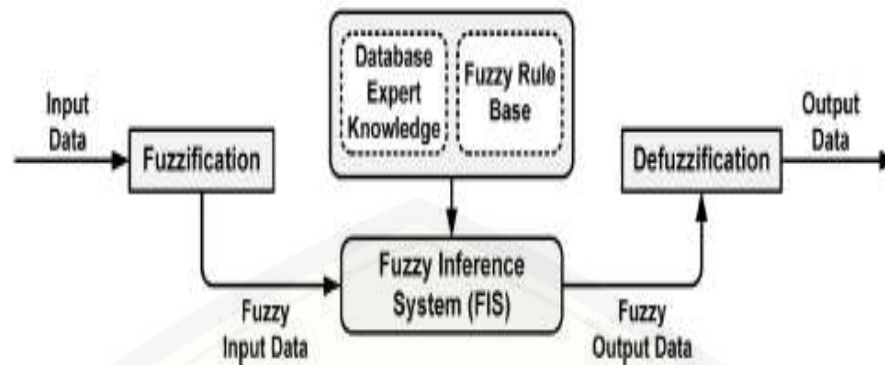
$$\mu_{\bar{A}} = 1 - \mu_A(X) \quad (2.13)$$

e. Operator-operator Alternatif

Pada dasarnya, ada 2 tipe operator alternatif, yaitu operator alternatif yang didasarkan pada transformasi aritmetika, seperti : mean, product dan bounded sum serta operator alternatif yang didasarkan pada transformasi fungsi yang lebih kompleks seperti Klas Yager dan Sugeno.

2.6.4 Fuzzy Inference System

FIS adalah metodologi sistem kontrol dengan merumuskan pemetaan keluaran untuk setiap masukan dengan menggunakan logika *fuzzy*. Sistem inferensi *fuzzy* berguna untuk menyelesaikan permasalahan dalam berbagai bidang industri, seperti kontrol otomatis, analisis keputusan, klasifikasi data, sistem pakar dan pengenalan pola (Alazzam dan Ahmed, 2019). Secara umum sistem *fuzzy* terdiri 3 tahap yaitu *fuzzification*, *fuzzy inference unit* dan *defuzzification*.



Gambar 2.15 Blok diagram sistem inferensi *fuzzy*

(sumber : Conker, 2019)

a. *Fuzzification*

Fuzzification atau fuzzifikasi adalah proses yang dilakukan dengan mengubah variabel *input* menjadi variabel *fuzzy* melalui pemetaan skala. Proses fuzzifikasi berfungsi untuk mempermudah pemetaan variabel *input* ke dalam himpunan *fuzzy*. Pemetaan dilakukan dengan dengan model *membership function* agar dapat diketahui besar masukan tersebut. *Membership function fuzzy* yang umum digunakan memiliki beberapa bentuk, yaitu :

- 1) Segitiga
- 2) Trapesium
- 3) Gaussian

b. *Fuzzy Inference Unit*

Inferensi *fuzzy* adalah proses penarikan kesimpulan berdasarkan nilai linguistik variabel *input* dan akan menghasilkan keputusan yang berupa nilai linguistik variabel *output*. Dalam proses inilah akan terjadi operasi himpunan *fuzzy* seperti operasi AND, OR dan implikasi.

1) *Fuzzy rule base*

Fuzzy Rule Base didefinisikan sebagai pengambil keputusan untuk membuat keputusan dirancang berdasarkan konsep *fuzzy* yaitu menggunakan kondisi IF-THEN untuk mendapatkan keluaran yang diinginkan.

2) *Knowledge base*

Memberikan informasi yang digunakan untuk menentukan kontrol linguistik memutuskan *membership function* berdasarkan fungsi mereka pengetahuan tentang sistem.

Terdapat tiga tipe sistem inferensi *fuzzy* yang digunakan dalam berbagai aplikasi. Perbedaan dari ketiga sistem inferensi ini adalah dalam konsekuensi aturan *fuzzy* dan proses defuzzifikasi.

1) Sistem Inferensi *Fuzzy* Mamdani

Metode Mamdani dikenal juga metode MAX-MIN. yang diperkenalkan oleh Ebrahim Mamdani pada tahun 1975. Basis aturan yang digunakan metode ini adalah sebagai berikut.

$$\text{Jika } (x_1 \text{ adalah } A_1) * (x_2 \text{ adalah } A_2) * \dots * (x_n \text{ adalah } A_n) \quad (2.14)$$

$$\text{Maka } (y \text{ adalah } B) \quad (2.15)$$

2) Sistem Inferensi *Fuzzy* Sugeno

Metode Sugeno diperkenalkan oleh Takagi Sugeno Kang pada tahun 1985. Metode Sugeno hampir sama dengan metode Mamdani perbedaannya terletak pada konsekuensi aturan *fuzzy*. Tidak seperti metode Mamdani yang konsekuensinya berbentuk aturan himpunan *fuzzy*, metode Sugeno memiliki konsekuensi berbentuk konstanta atau persamaan. Secara umum metode Sugeno adalah sebagai berikut.

$$\text{Jika } (x_1 \text{ adalah } A_1) * (x_2 \text{ adalah } A_2) * \dots * (x_n \text{ adalah } A_n) \quad (2.16)$$

$$\text{Maka } (y = f(x_1, x_2, \dots, x_n)) \quad (2.17)$$

Dengan $f(x_1, x_2, \dots, x_n)$ biasanya adalah *konstanta polinomial*. Contoh aturan untuk metode Sugeno adalah sebagai berikut.

$$\text{Jika } (x \text{ adalah 'Positif'}) \text{ maka } (y = 0.1x + 0.6) \quad (2.18)$$

$$\text{Jika } (x \text{ adalah 'nol'}) \text{ maka } (y = 0.2x + 0.7) \quad (2.19)$$

$$\text{Jika } (x \text{ adalah 'Positif'}) \text{ maka } (y = 0.2x + 0.8) \quad (2.20)$$

3) Sistem Inferensi *Fuzzy* Tsukamoto

Pada metode Tsukamoto, setiap konsekuensi pada aturan jika-maka ditunjukkan oleh himpunan *fuzzy* dengan *membership function* yang monoton. Keluarannya adalah nilai *crisp* yang diberikan pembobotan aturan dan keseluruhan

keluarannya adalah rata-rata pembebanan setiap keluaran aturan. Walaupun metode ini lebih cepat, namun metode ini jarang digunakan karena tidak transparan seperti metode yang lain.

c. Defuzzification

Defuzzification atau defuzzifikasi adalah proses mengubah hasil penerapan aturan dari bentuk logika *fuzzy* ke dalam bentuk himpunan *fuzzy* untuk mendapatkan kembali bentuk aslinya. Ada beberapa metode untuk defuzzifikasi tipe Mamdani seperti *center of area method* (COA) atau *Centroid, the largest of maximum* (LOM), *bisector of area, mean of maximum* (MOM) and *smallest of maximum* (SOM). Metode COA diskrit sering digunakan untuk defuzzifikasi karena mudah dalam mengimplementasikan dalam bahasa pemrograman. Persamaan defuzzifikasi COA dinyatakan dalam persamaan sebagai berikut: (Olajide dkk., 2017).

$$x^* = \frac{\sum_{i=1}^n \mu A(x) * (x)}{\sum_{i=1}^n \mu A(x)} \quad (2.21)$$

Keterangan :

x : Variabel keluaran

$\mu A(x)$: *Membership function* gabungan

x^* : Defuzzifikasi keluaran

n : Jumlah variabel

i : Variabel *counter*

2.7 Arduino DUE

Arduino DUE adalah *board microcontroller* pertama berbasis ARM 32-bit yakni ARM Cortex-M3 AT91SAM3X8EA. Arsitektur ARM atau *Advanced RISC Machine* adalah arsitektur prosesor 32 bit RISC yang dikembangkan oleh ARM Limited. Arduino DUE memiliki 54 pin masukan atau keluaran digital, dengan 12 pin dapat digunakan sebagai *output* PWM, 12 pin analog 12 bit, 4 pin UART, 2 pin DAC, 2 pin TWI, header SPI dan JTAG, 1 pin keluaran analog dengan *internal flash memory* sebesar 512kb, SRAM 96 kb dan kecepatan *clock frequency* sebesar 84 MHz. Arduino DUE bekerja pada tegangan 3,3V (Jalden dkk., 2018).



Gambar 2.16 Modul Arduino DUE

(Sumber : www.arduino.cc)

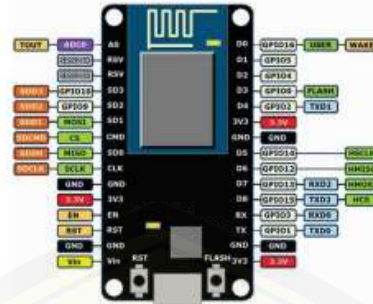
Arduino DUE sudah dilengkapi *bootloader* sehingga tidak memerlukan *flash program external* dan proses *upload* program menjadi lebih cepat dan sederhana. Untuk komunikasi dengan komputer dapat menggunakan micro-USB. Spesifikasi Arduino DUE dapat dilihat pada Tabel 2.1.

Tabel 2.1 Spesifikasi Arduino DUE

Item	Keterangan
Mikrokontroler	ARM Cortex-M3 AT91SAM3X8EA
Tegangan Operasi	3,3 V
Tegangan Masukan	7-12 V
Pin Digital I/O	54 pin
Pin Analog	12 pin
Arus DC per I/O	20 mA
Flash Memory	512 kb
SRAM	96 kb
Clock Frequency	84 MHz

2.8 NodeMCU ESP8266

Modul ESP8266 adalah modul Wi-Fi yang berfungsi sebagai perangkat pendukung mikrokontroler agar dapat terhubung langsung dengan Wi-Fi dan membuat koneksi TCP/IP. NodeMCU ESP8266 sudah terintegrasi dengan Arduino IDE yang sudah dilengkapi GPIO, I2C, UART, ADC, PWM dan Wi-Fi (Suprianto dan Wirawan, 2018). Spesifikasi modul NodeMCU ESP8266 dapat dilihat pada Tabel 2.2.



Gambar 2.17 Modul NodeMCU ESP8266

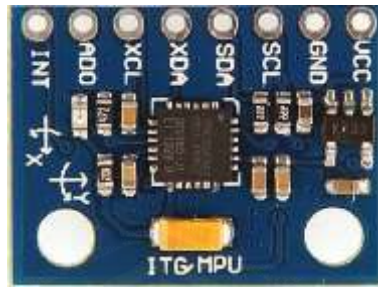
(Sumber : Wan, Song, and Cao 2019)

Tabel 2.2 Spesifikasi modul NodeMCU ESP8266

Item	Keterangan
Wi-Fi protocol	802.11 b/g/n
Input Voltage	3,0-3,3 VDC
Range Frequency	2,4 GHz
Flash Memory	4 MB

2.9 Sensor Sudut

Sensor MPU 6050 merupakan sensor yang terdiri dari *3-axis accelerometer* dan *3-axis gyroscope* dengan teknologi *micro electro mechanical system* (MEMS). Sensor *accelerometer* digunakan untuk mengukur percepatan, perubahan posisi dan percepatan gravitasi. Sensor *accelerometer* mendeteksi percepatan pada tiga sumbu yaitu sumbu x, sumbu y, dan sumbu z berdasarkan percepatan gravitasi bumi. Percepatan bumi diukur dengan satuan *g-force* (G), untuk 1G sama dengan $9,8 \text{ m/s}^2$. Sensor *gyroscope* digunakan untuk mengukur kecepatan sudut benda yang bergerak terhadap sumbu putar. Sensor *gyroscope* menghasilkan sinyal yang nilainya sebanding dengan kecepatan sudut tegak lurus terhadap sumbu putar.



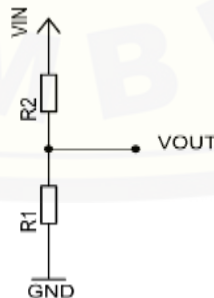
Gambar 2.18 Modul sensor GY-521 MPU 6050

(Sumber : Ambildhok and Hulle 2017)

Satuan kecepatan sudut dinyatakan dalam radian per detik (rad/s) atau derajat per detik ($^{\circ}/s$) (Amirullah dkk., 2019). Sensor GY-521 MPU 6050 merupakan salah satu jenis sensor MPU 6050. Sistem komunikasi yang digunakan I²C dengan tegangan kerja 3,3-5 Volt (Zeng dkk., 2018).

2.10 Sensor Tegangan

Prinsip pembagi tegangan atau *voltage divider* yang digunakan untuk menurunkan tegangan agar dapat dibaca oleh analog pin arduino. Tegangan merupakan besaran analog sehingga dalam dunia analog dapat diproses langsung. Di dunia digital besaran analog dapat dikonversi menjadi besaran diskrit dengan ADC (*Analog to Digital Converter*) atau sebaliknya dengan DAC (*Digital to Analog Converter*).



Gambar 2.19 Resistor sebagai pembagi tegangan

(Sumber : Ismail 2019)

Persamaan matematis pembagi tegangan untuk sensor tegangan sebagai berikut :

$$V_{out} = \frac{R_2}{R_1 + R_2} \times V_{in} \quad (2.22)$$

Tegangan maksimum yang dapat dibaca arduino 5 Volt sehingga V_{out} harus kurang dari 5 Volt dengan pengaturan resistansi R_1 dan R_2 agar tidak melebihi 5 Volt (Ghodake dkk., 2019).

2.11 Sensor Arus

Sensor ACS712 merupakan modul yang digunakan untuk mendeteksi arus yang mengalir dengan menggunakan *current sensor chip* ACS712 yang memanfaatkan efek *hall effect* untuk memberikan aliran listrik atau elektron yang dipengaruhi medan magnet. Sensor arus ACS712 juga sudah terintegrasi dengan Arduino IDE. Arus maksimum yang dapat dideteksi sebesar 20A, dengan tegangan pada pin keluaran akan berubah secara linier mulai dari 2,5V. Perubahan tingkat tegangan secara linier terhadap besar arus sebesar 100 mV/ Ampere.



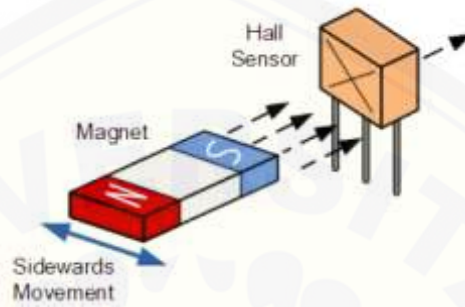
Gambar 2.20 Modul sensor ACS712

(Sumber : Gumelar B and Ariyanto 2017)

2.12 Sensor Kecepatan

Sensor kecepatan menggunakan prinsip kerja dari sensor *hall effect* yang mendeteksi medan magnet, maka dapat digunakan sebagai sensor kecepatan. Ketika sensor *hall effect* mendeteksi adanya medan magnet, sensor akan mengirimkan

sinyal dan sebaliknya. Magnet yang diletakan di bagian putaran roda akan dideteksi satu kali oleh sensor *hall effect* untuk satu kali putaran penuh , oleh itu waktu yang diperlukan untuk sensor *hall effect* dalam berputar untuk mendeteksi medan magnet yang akan digunakan untuk mengukur kecepatan putaran.



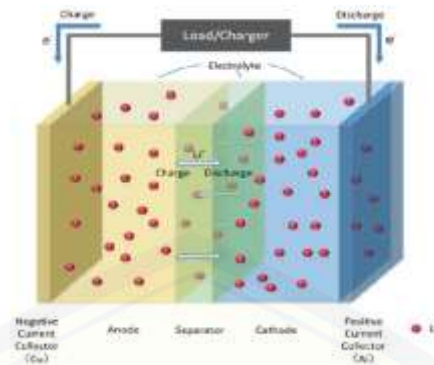
Gambar 2.21 Mekanisme penempatan sensor *hall effect*

(Sumber : Ismail 2019)

Sensor *hall effect* dipasang di depan perputaran roda yang berhadapan lurus dengan magnet. Sensor *hall effect* akan menghitung putaran roda, jika sensor *hall effect* mendeteksi magnet akan mengirimkan sinyal *high* dan apabila tidak mendeteksi magnet akan mengirimkan sinyal *low*.

2.13 Baterai *Lithium Ion*

Baterai *lithium Ion* merupakan jenis baterai yang dapat diisi ulang (*rechargeable battery*), dengan *ion lithium* bergerak dari katoda ke anoda selama proses pengisian (*charging*) dan mengalami hal sebaliknya ketika pelepasan (*discharging*). Elemen baterai *lithium ion* sendiri terdiri anoda, katoda dan pemisah antara anoda dan katoda (*separator*). Anoda berasal dari karbon grafit dengan beberapa pengikat dan konduktif adiktif. Katoda berasal dari senyawa yang dilapisi *aluminium foil*. Lembaran berpori tipis digunakan sebagai isolasi keduanya.

Gambar 2.22 Elemen baterai *lithium ion*

(Sumber : J. Zhang et al. 2018)

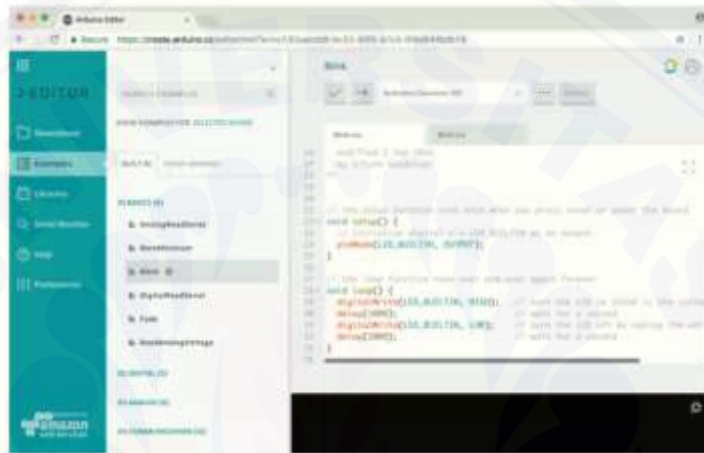
Penggunaan baterai *lithium ion* sangat populer saat ini sebagai sistem daya seperti *smartphone*, kendaraan listrik, satelit dan peralatan lainnya dikarenakan memiliki kepadatan energi, tingkat *self-discharge* rendah, tegangan *discharge* tinggi dan siklus hidup tinggi (Peng dkk., 2019). Kepadatan energi baterai *lithium ion* lebih besar dua kali lipat dari baterai *nickel-cadmium*. Dibandingkan dengan baterai *nickel-cadmium* tegangan per *cell* baterai *lithium ion* sebesar 3,6V sementara baterai *nickel-cadmium* sebesar 1,2 V. Baterai *lithium ion* memerlukan *electronically circuit protection external* untuk mengamankan baterai baik saat pengisian maupun saat pelepasan ion lithium (Ismail, 2019). Kelebihan dan kekurangan baterai *lithium ion* dapat dilihat pada Tabel 2.3.

Tabel 2.3 Kelebihan dan kekurangan baterai *lithium ion*

Kelebihan	Kekurangan
Energy density lebih besar	Membutuhkan <i>electronically circuit protection</i>
Arus <i>discharge</i> besar	Lebih mahal
<i>Low self-discharge</i>	Komposisi bahan kimia dapat berubah pada kondisi tertentu
<i>Low maintenance</i>	Proses distribusi yang rumit ketat ketika jumlah besar

2.14 Software Arduino IDE

IDE atau *Integrated Development Environment* adalah program komputer yang digunakan untuk pengembangan perangkat lunak. Arduino IDE merupakan aplikasi lintas platform (Windows, mac OS X dan Linux) ditulis dengan bahasa java dan didasarkan pada pemrosesan dan perangkat lunak *open-source* lainnya yang memudahkan dalam menulis dan mengunggah *sketch* ke board arduino.



Gambar 2.23 Tampilan Arduino IDE

(Sumber : www.Arduino.cc)

Software Arduino IDE terdiri dari tiga bagian yaitu :

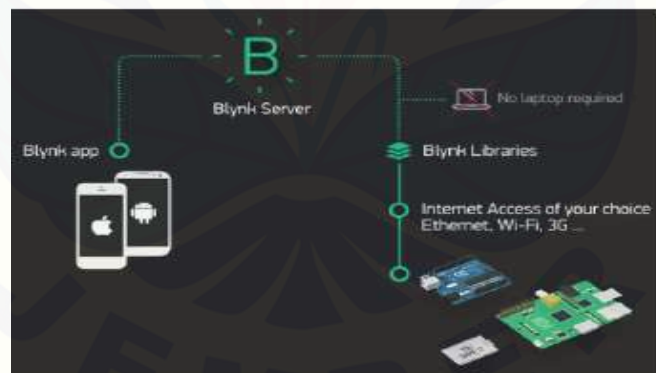
1. *Editor Program*, digunakan untuk menulis dan mengedit program atau *sketch*.
2. *Compiler*, digunakan untuk mengubah sketch menjadi kode biner. Karena kode biner bahasa yang dipahami mikrokontroler.
3. *Uploader*, digunakan untuk mengunggah sketch ke papan mikrokontroler.

Struktur perintah pada Arduino IDE secara garis besar terdiri dari *void setup* dan *void loop*. *Void setup* sendiri berisi perintah yang akan dieksekusi hanya satu kali saat arduino dihidupkan sedangkan *void loop* berisi perintah yang akan dieksekusi berulang-ulang selama arduino dinyalakan (Ismail, 2019).

2.15 Aplikasi Blynk

Blynk merupakan platform aplikasi yang dapat diunduh secara gratis untuk sistem operasi iOS dan Android. Blynk memungkinkan pengguna untuk terhubung dengan perangkat *Internet of Things* (IoT) sehingga dapat mengendalikan perangkat keras dari jarak jauh, dapat menampilkan data sensor, menyimpan data dan memvisualisasikan data kepada pengguna. Blynk dapat diimplementasikan pada mikrokontroler Arduino, Raspberry PI, Wemos D1 mini yang terhubung internet (Waworundeng dkk., 2019). Terdapat tiga komponen utama Blynk yaitu :

1. Aplikasi Blynk yang digunakan untuk mengontrol perangkat dan menampilkan data pada *widget*.
2. Blynk *server* yang merupakan layanan *cloud* yang bertanggung jawab untuk semua komunikasi *smartphone* dengan *hardware*.
3. Blynk *library* mencakup berbagai widget seperti tombol kontrol, format tampilan, notifikasi dan manajemen waktu yang memungkinkan perangkat dapat mengirim data sensor yang ditampilkan di *smartphone*.



Gambar 2.24 Arsitektur *cloud* Blynk

(Sumber : Serikul, Nakpong, and Nakjuatong 2019)

BAB 3. METODE PENELITIAN

3.1 Tempat dan Waktu Penelitian

3.1.1 Tempat Penelitian

Penelitian ini dilakukan di Laboratorium Sistem Tenaga dan sekretariat Tim Mobil Listrik TITEN, Universitas Jember.

3.1.2 Waktu Penelitian

Waktu penelitian dilaksanakan selama kurang lebih 6 bulan, dapat dilihat pada Tabel 3.1.

Tabel 3.1 Jadwal waktu penelitian

No	Kegiatan	Bulan				
		1	2	3	4	5
1	Studi literatur	■	■	■	■	
2	Perancangan dan pembuatan alat		■	■	■	
3	Pengujian alat		■	■	■	
4	Analisa dan pengambilan data			■	■	■
5	Penyusunan laporan				■	■
6	Ujian					■

Keterangan :

■ : Kegiatan dilaksanakan

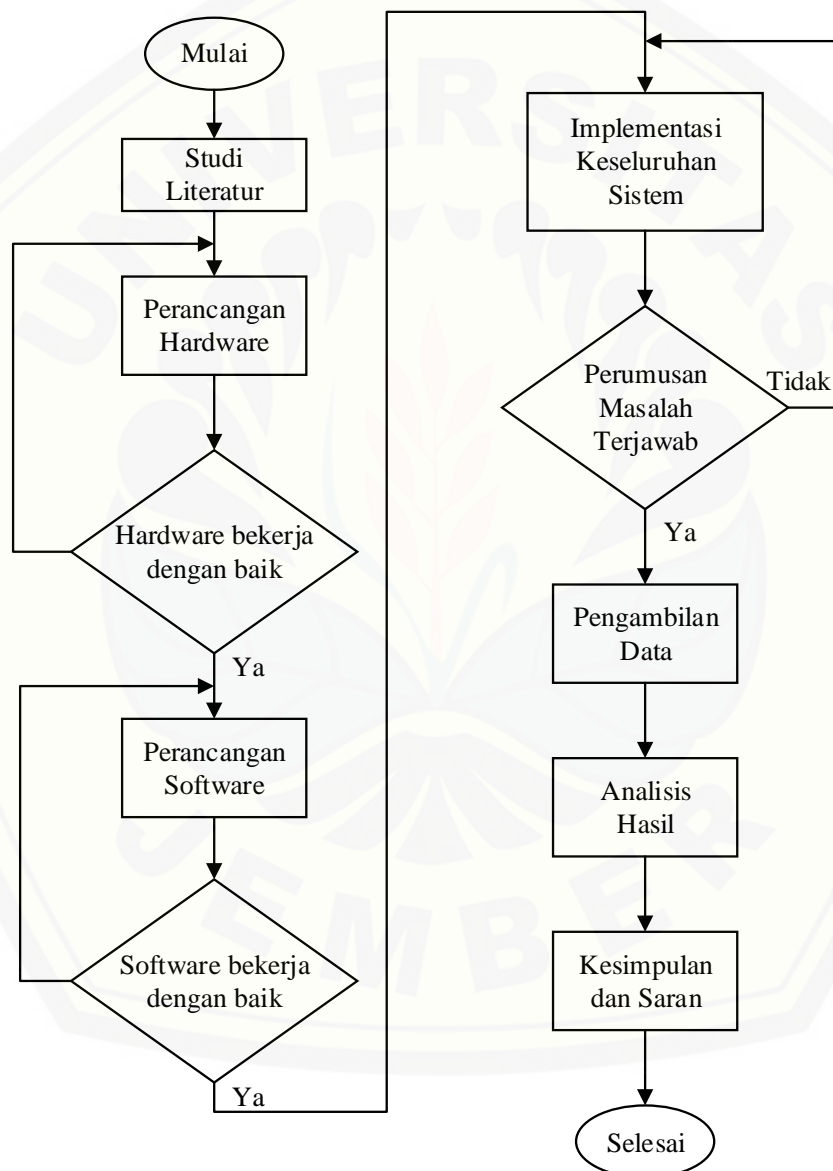
3.2 Alat dan Bahan Penelitian

Alat dan bahan yang diperlukan pada penelitian ini adalah sebagai berikut:

- 1) Laptop/PC
- 2) Ponsel pintar
- 3) Arduino DUE
- 4) Motor BLDC 350 Watt
- 5) Baterai *Lithium Ion* 36 Volt
- 6) Komponen elektronika
- 7) Alat ukur : AVO meter, *Tachometer*

- 8) Modul sensor : ACS712, *hall effect*, *voltage divider*. dan MPU 6050
- 9) *Software* : Microsoft Word 2013, Microsoft Excell 2016, Matlab 2015, Arduino IDE dan Blynk

3.3 Tahap Penelitian



Gambar 3.1 Flowchart penelitian

Pada proses penelitian ini terdapat beberapa tahapan penelitian, berikut adalah langkah-langkah penelitian meliputi :

a. Studi literatur

Pada tahap ini merupakan tahapan mencari informasi yang berkaitan dengan penelitian yang akan dilakukan, yaitu jurnal yang berkaitan dengan sistem *monitoring* dan kontrol berbasis *Internet of Things* (IoT) yang berasal dari jurnal nasional maupun internasional. Diharapkan dengan informasi yang didapat sebagai acuan dalam penelitian sehingga sesuai yang diharapkan.

b. Perancangan dan pembuatan alat

Pada tahap ini dilakukan desain alat dengan *software Eagle* atau *Easily Applicable Graphical Layout Editor* versi 9.0.1 dengan acuan dari informasi yang sudah didapatkan sebelumnya. Perancangan alat diawali dengan desain skematik dengan merangkai komponen kemudian desain pada board untuk dicetak di PCB. Pembuatan desain sistem *monitoring* terdiri dari beberapa sensor yang disimpan pada data logger. Desain sistem kontrol FIS menggunakan *software* Matlab 2015 yang kemudian dikonversi menjadi program FIS Arduino.

c. Pengujian alat

Pada tahap ini dilakukan pengujian alat untuk mengetahui apakah alat dapat bekerja dengan baik dan sesuai dengan penelitian atau tidak. Pengujian terdiri dari pengujian dan kalibrasi sensor, pengujian *Graphic User Interface* Blynk, pengujian *monitoring*, pengujian PWM dan pengujian kontrol *fuzzy*. Jika alat sudah sesuai dengan penelitian maka akan dilanjutkan tahapan analisa dan pengambilan data.

d. Analisa dan pengambilan data

Pada tahap ini dilakukan pengambilan data alat dengan beberapa pengujian dan selanjutnya menganalisa data yang sudah didapatkan. Pengambilan data sistem *monitoring* dilakukan secara dinamis sejauh 2 KM dan beberapa kondisi jalan sedangkan untuk sistem kontrol mobil listrik dengan mengendalikan kecepatan saat kondisi jalan tanjakan dan turunan serta penggunaan energi menggunakan FLC dan tanpa FLC

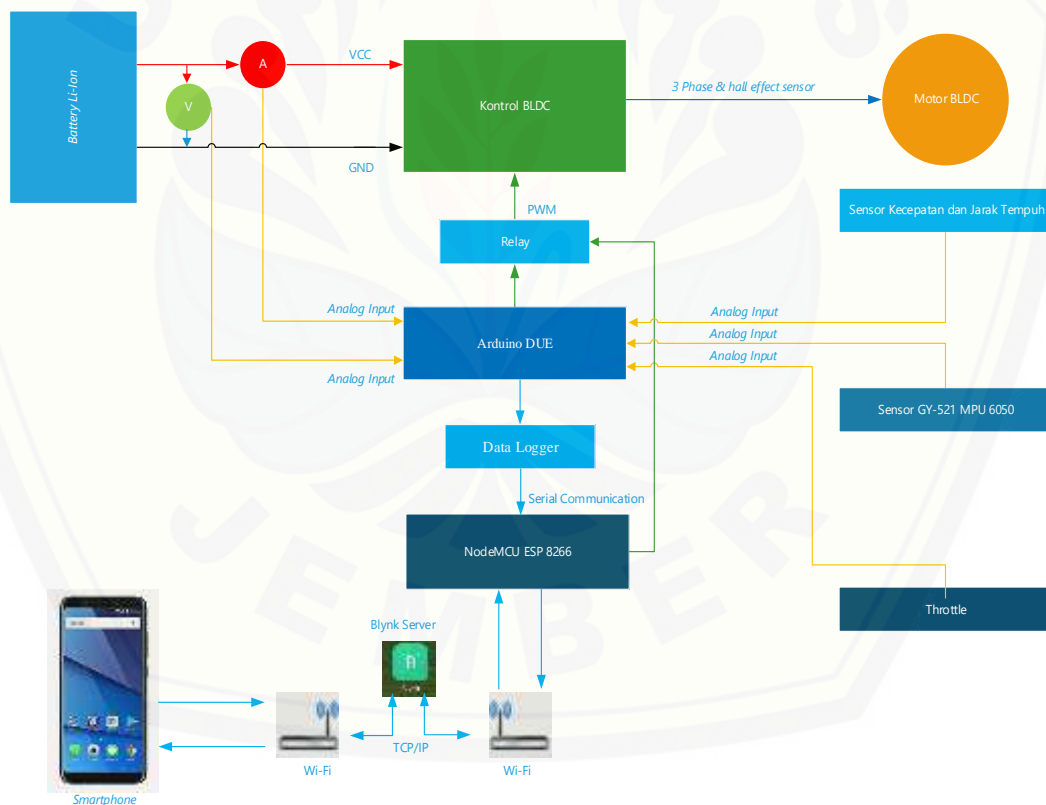
e. Penyusunan laporan

Pada tahap ini dilakukan penyusunan laporan dari pengambilan data alat dan analisa yang sudah didapatkan ke pembahasan. Kemudian menyimpulkan dari penelitian yang telah dilakukan dan memberikan saran untuk memperbaiki kekurangan pada saat penelitian.

3.4 Perancangan Alat

Penelitian ini, perancangan alat tersusun atas perancangan blok diagram sistem, perancangan *wiring* diagram sistem, perancangan sistem FLC dan *flowcart* keseluruhan sistem.

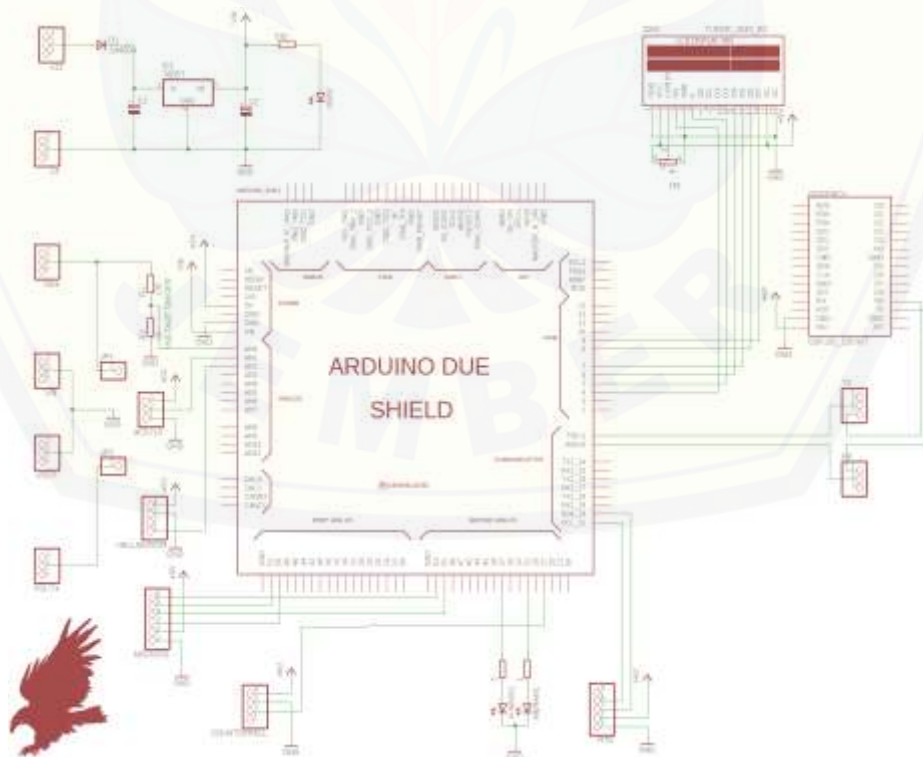
3.4.1 Perancangan Blok Diagram Sistem



Gambar 3.2 Blok diagram sistem

Baterai *Lithium Ion* 36V 10Ah merupakan sumber utama mobil listrik dengan motor BLDC 250W sebagai penggerak utama. Pengendali kecepatan motor BLDC menggunakan buatan pabrik yaitu *sunrace* 500W. Kecepatan mobil listrik dapat dikendalikan melalui pedal gas yang berasal dari potensiometer. Sensor tegangan dipasang pada baterai Li-Ion 36V dengan tegangan *input* maksimal 42V. Sensor arus dipasang secara seri antara baterai Li-Ion dengan beban yang masuk ke sistem pengendali dengan batas maksimal 20A. Sensor kecepatan dan sensor jarak tempuh dipasang di pinggir roda belakang untuk mendeteksi magnet yang terpasang di roda. Sensor sudut dipasang sejajar dengan rangka depan mobil listrik. Relay digunakan untuk menghubungkan PWM Arduino DUE ke *driver* MOSFET kontrol BLDC berdasarkan *mode driver* ataupun *mode* FLC. Data sensor akan diolah oleh Arduino DUE dan disimpan ke data logger. NodeMCU ESP8266 digunakan untuk mengirim data sensor ke GUI Blynk.

3.4.2 Perancangan Skematik Sistem



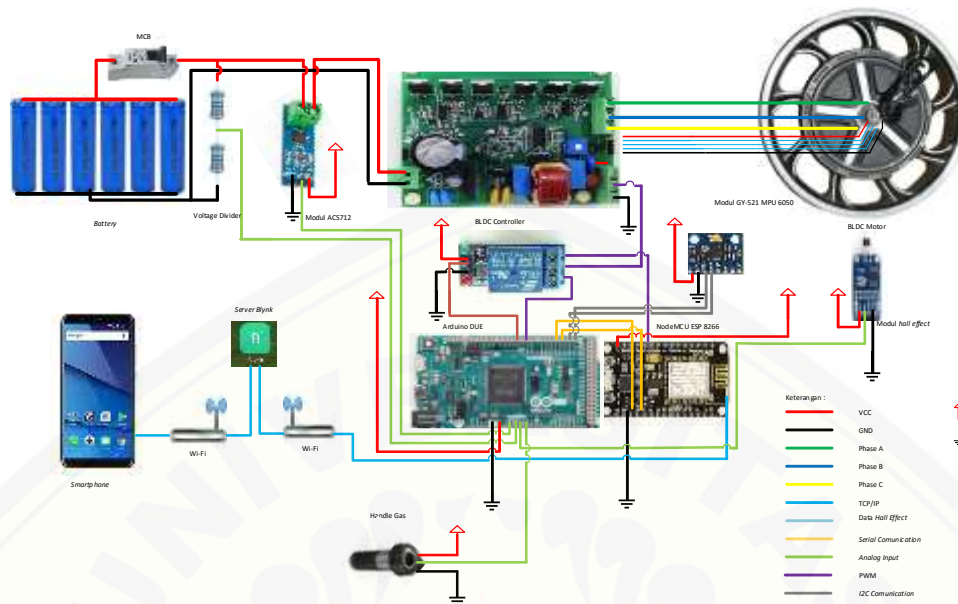
Gambar 3.3 Skematik sistem

Perancangan skematik menggunakan *software Eagle* atau *Easily Applicable Graphical Layout Editor* versi 9.0.1. *Software Eagle* merupakan *software* gratis yang dapat digunakan untuk mendesain skematik elektronik dan PCB atau *Printed Circuit Board*. Perancangan skematik ini terdiri dari beberapa komponen elektronika yaitu Arduino DUE, NodeMCU ESP8266, sensor tegangan, sensor arus, sensor kecepatan, sensor sudut dan sensor jarak tempuh, SD card, *Real Time Clock*, *Liquid Crystal Display* (16x2) dan *Light Emitting Diode*.

Sensor tegangan menggunakan dua buah resistor dengan prinsip pembagi tegangan yaitu R1 sebesar 10 K Ω dan R2 sebesar 1 K Ω sehingga tegangan yang masuk ke Arduino DUE kurang dari 5V. Sensor arus menggunakan modul ACS712 yang dapat membaca arus maksimal yang mengalir sebesar 20A. Sensor kecepatan dan jarak tempuh menggunakan sensor *hall effect* 319 dengan mendeteksi adanya magnet yang kemudian dikonversi menjadi putaran per menit (RPM) dan jumlah putaran roda yang digunakan sebagai sensor jarak tempuh. Sensor sudut menggunakan sensor MPU 6050 untuk mendeteksi sudut sumbu Y.

Data sensor tegangan, arus, sudut, kecepatan, dan jarak tempuh terhubung dengan pin analog mikrokontroler. Sensor sudut dan RTC terhubung dengan Arduino DUE melalui komunikasi I2C. Modul SD Card terhubung dengan Arduino DUE melalui komunikasi SPI. Arduino DUE dan NodeMCU ESP8266 saling terhubung melalui pin Rx dan Tx yaitu dengan komunikasi serial. Arduino DUE digunakan untuk mengubah data analog sensor menjadi data digital. Data sensor disimpan di SD card kemudian dikirim ke NodeMCU ESP8266. NodeMCU ESP8266 digunakan untuk mengirimkan data dari Arduino DUE ke GUI Blynk melalui jaringan internet.

3.4.3 Perancangan *Wiring* Diagram Sistem



Gambar 3 4 Wiring diagram sistem

Wiring diagram sistem digunakan untuk menggambarkan secara keseluruhan sistem *monitoring* dan kontrol kecepatan mobil listrik. Sistem *monitoring* digunakan untuk *monitoring* tegangan, arus, daya, energi, sudut, pedal gas, PWM, kecepatan, dan jarak tempuh mobil listrik TITEN EV-3. TITEN EV-3 merupakan mobil listrik kategori *Urban Concept* atau mobil perkotaan dengan satu pengemudi dengan sumber tegangan baterai Li-Ion 36V 10Ah menggunakan motor BLDC 250W sebagai penggerak utama.

Pengendali kecepatan Motor BLDC menggunakan pengendali kecepatan buatan pabrik yaitu *sunrace* 500W. Kecepatan mobil listrik dapat dikendalikan melalui pedal gas yang berasal dari potensiometer. Sensor tegangan dipasang pada baterai Li-Ion 36V dengan tegangan *input* maksimal 42V. Sensor arus dipasang secara seri antara baterai Li-Ion dengan beban yang masuk ke sistem pengendali dengan batas maksimal 20A. Sensor kecepatan dan sensor jarak tempuh dipasang di pinggir roda belakang untuk mendeteksi magnet yang terpasang di roda. Sensor sudut dipasang sejajar dengan rangka depan mobil listrik. Data sensor akan diolah

oleh Arduino DUE dan disimpan ke data logger. NodeMCU ESP8266 digunakan untuk mengirim data sensor ke GUI Blynk.

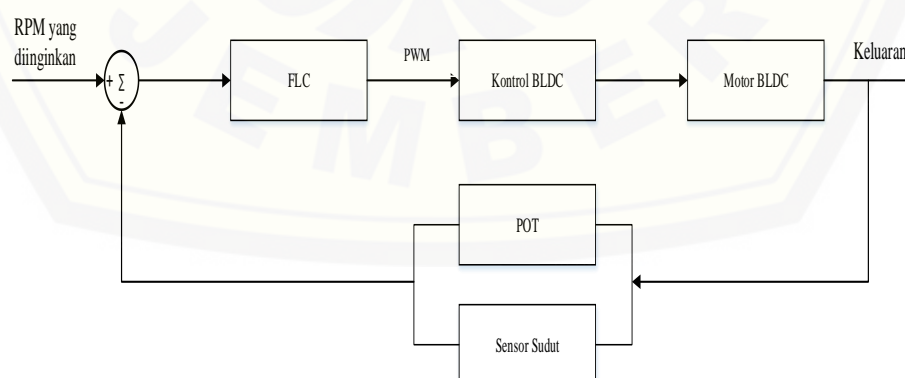
3.4.4 Perancangan Sistem FLC

Sistem *fuzzy* bekerja dengan himpunan *fuzzy* dan operasi himpunan *fuzzy*. Himpunan *fuzzy* digunakan mempresentasikan nilai linguistik suatu variabel untuk memperluas jangkauan fungsi karakteristik sehingga fungsi tersebut dapat mencakup bilangan *real* pada *interval* antara 0 dan 1. Tipe Mamdani menyediakan struktur yang relatif sederhana untuk operasi “MIN-MAX” oleh karena itu, merupakan salah satu aplikasi yang paling umum dapat diterapkan dalam segala jenis masalah yang memiliki *input* dan *output* yang saling berhubungan satu sama lain (Conker, 2019).

Sistem *fuzzy* digunakan manajemen kinerja kontrol motor BLDC saat di tanjakan dan turunan dengan tipe Mamdani. Variabel *input* yang digunakan yaitu POT dan sudut, serta variabel *output* yaitu PWM. Perancangan kontrol *fuzzy* terdiri dari beberapa tahapan, yaitu *fuzzification*, *fuzzy inference*, dan *defuzzification*.

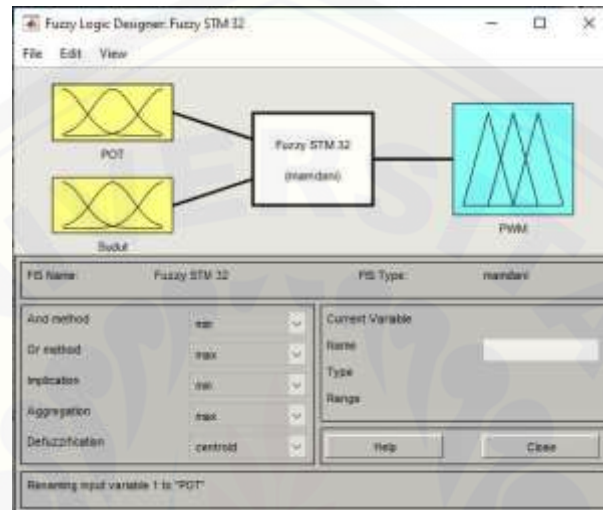
a. *Fuzzification*

Pada tahap ini proses mengubah nilai variabel numerik ke nilai variabel linguistik atau dengan kata lain pemetaan dari ruang *input* ke himpunan *fuzzy* yang didefinisikan pada semesta pembacaan variabel *input*.



Gambar 3.5 Diagram sistem kontrol *fuzzy close-loop*

Gambar 3.5 merupakan sebuah sistem FLC *close-loop* yang digunakan untuk mengendalikan kecepatan mobil listrik berdasarkan kondisi jalan tanjakan dan turunan. Variabel *input* FLC merupakan nilai ADC potensiometer dan sudut terukur dari sensor MPU 6050.



Gambar 3.6 Fuzzy logic designer pada Matlab

Gambar 3.6 menunjukkan variabel *input* dan *output* yang menggunakan *fuzzy logic controller* tipe Mamdani. Variabel *input* berupa nilai ADC potensiometer dan sudut yang kemudian diproses dengan aturan IF-THEN sesuai aturan *rule-base* yang telah dibuat. *Membership function* berjumlah tujuh diharapkan memiliki osilasi yang lebih halus dengan *set-point*. Variabel *output* berupa PWM dari Arduino DUE.

POT = Nilai ADC potensiometer dengan range 0 sampai 255

Sudut = Sudut terbaca sensor dengan range -60° sampai 60°

PWM = *Output* digital Arduino DUE dengan range 0 sampai 255

Berikut parameter variabel *input* dan *output* pada penelitian sebagai berikut

:

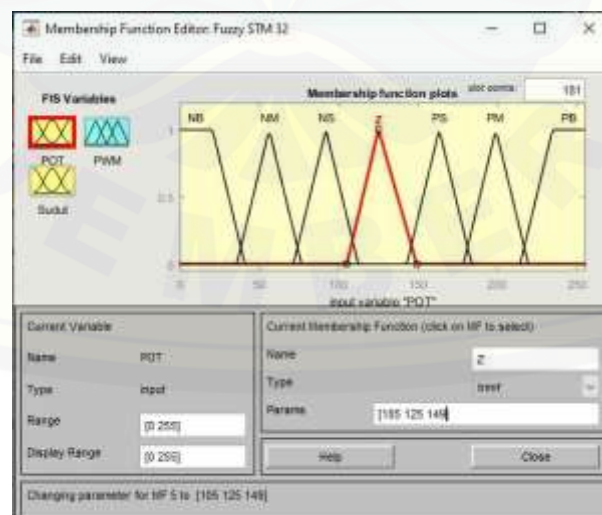
1) Parameter POT

Adapun parameter POT yang digunakan dalam kontrol *fuzzy* ini dapat dilihat pada Tabel 3.2.

Tabel 3.2 Himpunan Parameter POT

Parameter	Nilai
<i>Negative Big</i> (Nb)	0 s.d 40
<i>Negative Medium</i> (Nm)	35 s.d 75
<i>Negative Small</i> (Ns)	70 s.d 110
<i>Zero</i> (Z)	105 s.d 145
<i>Positive Small</i> (Ps)	140 s.d 180
<i>Positive Medium</i> (Pm)	175 s.d 215
<i>Positive Big</i> (Pb)	210 s.d 255

Berdasarkan Tabel 3.2 ditentukan tujuh himpunan *fuzzy* yang akan digunakan yaitu *Negative Big* (Nb), *Negative Medium* (Nm), *Negative Small* (Ns), *Zero* (Z), *Positive Small* (Ps), *Positive Medium* (Pm) dan *Positive Big* (Pb). Gambar 3.7 merupakan *membership function* yang digunakan.

Gambar 3.7 *Membership function* variabel POT

Berikut adalah *membership function* yang digunakan untuk variabel POT:

$$\mu (Nb) = \begin{cases} 0 & ; & x \leq 0 \text{ atau } x \geq 40 \\ (x - 0) / (20-0) & ; & 0 \leq x \leq 20 \\ 1 & ; & 20 \leq x \leq 30 \\ (40 - x) / (40 - 30) & ; & x \geq 40 \end{cases}$$

$$\mu (Nm) = \begin{cases} 0 & ; & x \leq 35 \text{ atau } x \geq 75 \\ (x - 35) / (55 - 35) & ; & 35 \leq x \leq 55 \\ (55 - x) / (75 - 55) & ; & 55 \leq x \leq 75 \end{cases}$$

$$\mu (Ns) = \begin{cases} 0 & ; & x \leq 70 \text{ atau } x \geq 110 \\ (x - 70) / (90 - 70) & ; & 70 \leq x \leq 90 \\ (90 - x) / (110 - 90) & ; & 90 \leq x \leq 110 \end{cases}$$

$$\mu (Z) = \begin{cases} 0 & ; & x \leq 105 \text{ atau } x \geq 145 \\ (x - 105) / (125 - 105) & ; & 105 \leq x \leq 125 \\ (125 - x) / (145 - 105) & ; & 125 \leq x \leq 145 \end{cases}$$

$$\mu (Ps) = \begin{cases} 0 & ; & x \leq 140 \text{ atau } x \geq 180 \\ (x - 140) / (160 - 140) & ; & 140 \leq x \leq 160 \\ (160 - x) / (180 - 160) & ; & 160 \leq x \leq 180 \end{cases}$$

$$\mu (Pm) = \begin{cases} 0 & ; & x \leq 175 \text{ atau } x \geq 215 \\ (x - 175) / (195 - 175) & ; & 175 \leq x \leq 195 \\ (195 - x) / (215 - 195) & ; & 195 \leq x \leq 215 \end{cases}$$

$$\mu (Pb) = \begin{cases} 0 & ; & x \leq 210 \text{ atau } x \geq 255 \\ (x - 210) / (230 - 210) & ; & 210 \leq x \leq 230 \\ 1 & ; & 230 \leq x \leq 240 \\ (255 - x) / (240 - 230) & ; & x \geq 255 \end{cases}$$

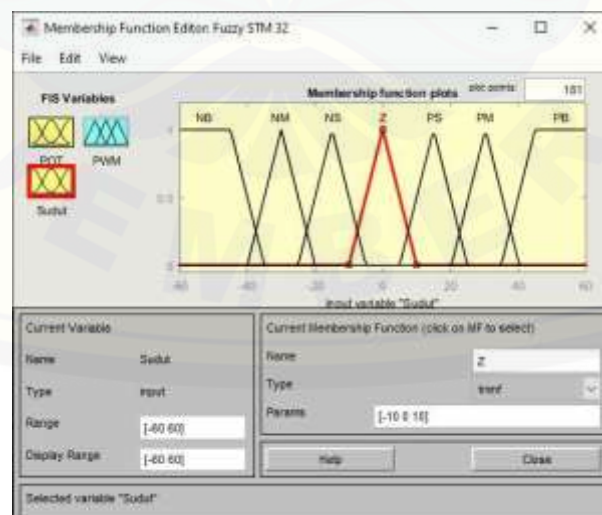
2) Parameter Sudut

Adapun parameter sudut yang digunakan dalam kontrol *fuzzy* dapat dilihat pada Tabel 3.3.

Tabel 3.3 Himpunan parameter sudut

Parameter	Nilai
<i>Negative Big</i> (Nb)	(-60) s.d (-35)
<i>Negative Medium</i> (Nm)	(-40) s.d (-20)
<i>Negative Small</i> (Ns)	(-25) s.d (-5)
<i>Zero</i> (Z)	(-10) s.d 10
<i>Positive Small</i> (Ps)	5 s.d 25
<i>Positive Medium</i> (Pm)	20 s.d 40
<i>Positive Big</i> (Pb)	35 s.d 60

Berdasarkan Tabel 3.3 ditentukan tujuh himpunan yang akan digunakan yaitu *Negative Big* (Nb), *Negative Medium* (Nm), *Negative Small* (Ns), *Zero* (Z), *Positive Small* (Ps), *Positive Medium* (Pm) dan *Positive Big* (Pb). Gambar 3.8 merupakan *membership function* yang digunakan.



Gambar 3.8 *Membership function* variabel sudut

Berikut adalah *membership function* yang digunakan untuk variabel sudut:

$$\mu (Nb) = \begin{cases} 0 & ; & x \leq -60 \text{ atau } x \geq -35 \\ (x - (-60)) / ((-50) - (-60)) & ; & -60 \leq x \leq -50 \\ 1 & ; & -50 \leq x \leq -45 \\ ((-35) - x) / ((-35) - (-45)) & ; & x \geq -35 \end{cases}$$

$$\mu (Nm) = \begin{cases} 0 & ; & x \leq -40 \text{ atau } x \geq -20 \\ (x - (-40)) / ((-30) - (-40)) & ; & -40 \leq x \leq -30 \\ ((-30) - x) / ((-20) - (-30)) & ; & -30 \leq x \leq -20 \end{cases}$$

$$\mu (Ns) = \begin{cases} 0 & ; & x \leq -25 \text{ atau } x \geq -5 \\ (x - (-25)) / ((-15) - (-25)) & ; & -25 \leq x \leq -15 \\ ((-15) - x) / ((-5) - (-15)) & ; & -15 \leq x \leq -5 \end{cases}$$

$$\mu (Z) = \begin{cases} 0 & ; & x \leq -10 \text{ atau } x \geq 10 \\ (x - (-10)) / (0 - (-10)) & ; & -10 \leq x \leq 0 \\ (0 - x) / (10 - 0) & ; & 0 \leq x \leq 10 \end{cases}$$

$$\mu (Ps) = \begin{cases} 0 & ; & x \leq 5 \text{ atau } x \geq 25 \\ (x - 5) / (15 - 5) & ; & 5 \leq x \leq 15 \\ (15 - x) / (25 - 15) & ; & 15 \leq x \leq 25 \end{cases}$$

$$\mu (Pm) = \begin{cases} 0 & ; & x \leq 20 \text{ atau } x \geq 40 \\ (x - 20) / (30 - 20) & ; & 20 \leq x \leq 30 \\ (30 - x) / (40 - 30) & ; & 30 \leq x \leq 40 \end{cases}$$

$$\mu (Pb) = \begin{cases} 0 & ; & x \leq 35 \text{ atau } x \geq 60 \\ (x - 35) / (45 - 35) & ; & 35 \leq x \leq 45 \\ 1 & ; & 45 \leq x \leq 50 \\ (60 - x) / (60 - 50) & ; & x \geq 60 \end{cases}$$

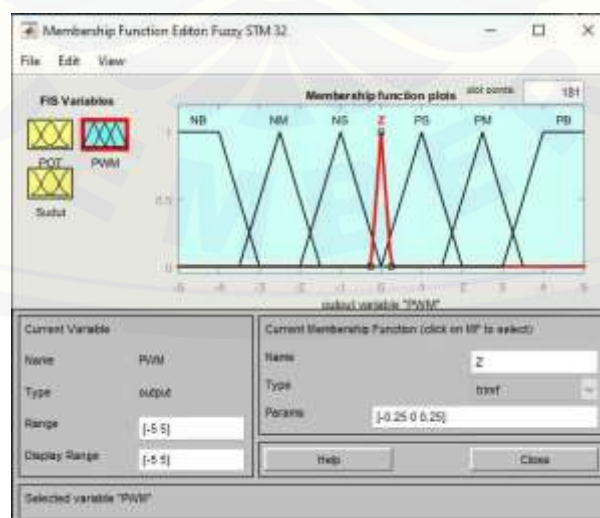
3) Parameter *Output*

Pada bagian ini merupakan kontrol *fuzzy* dengan *output* yang dipengaruhi *input* yang telah didapatkan. Keluaran dari *fuzzy* ini yakni berupa PWM yang didapatkan setelah pengambilan keputusan *rule-base* yang telah dibuat.

Tabel 3.4 Himpunan parameter PWM

Parameter	Nilai
<i>Negative Big</i> (Nb)	(-6) s.d (-3)
<i>Negative Medium</i> (Nm)	(-3,5) s.d (-1,5)
<i>Negative Small</i> (Ns)	(-2) s.d 0
<i>Zero</i> (Z)	(-0,25) s.d 0,25
<i>Positive Small</i> (Ps)	0 s.d 20
<i>Positive Medium</i> (Pm)	1,5 s.d 3,5
<i>Positive Big</i> (Pb)	3 s.d 6

Berdasarkan Tabel 3.4 ditentukan tujuh himpunan yang akan digunakan yaitu *Negative Big* (Nb), *Negative Medium* (Nm), *Negative Small* (Ns), *Zero* (Z), *Positive Small* (Ps), *Positive Medium* (Pm) dan *Positive Big* (Pb). Gambar 3.9 merupakan *membership function* yang digunakan.



Gambar 3.9 *Membership function* variabel PWM

Berikut adalah *membership function* yang digunakan untuk variabel PWM:

$$\mu (Nb) = \begin{cases} 0 & ; & x \leq -6 \text{ atau } x \geq -3 \\ (x - (-6)) / ((-5) - (-6)) & ; & -6 \leq x \leq -5 \\ 1 & ; & -5 \leq x \leq -4 \\ ((-3) - x) / ((-3) - (-4)) & ; & x \geq -3 \end{cases}$$

$$\mu (Nm) = \begin{cases} 0 & ; & x \leq -3,5 \text{ atau } x \geq -1,5 \\ (x - (-3,5)) / ((-2,5) - (-3,5)) & ; & -3,5 \leq x \leq -2,5 \\ ((-2,5) - x) / ((-1,5) - (-2,5)) & ; & -2,5 \leq x \leq -1,5 \end{cases}$$

$$\mu (Ns) = \begin{cases} 0 & ; & x \leq -2 \text{ atau } x \geq 0 \\ (x - (-2)) / ((-1) - (-2)) & ; & -2 \leq x \leq -1 \\ ((-1) - x) / (0 - (-1)) & ; & -1 \leq x \leq 0 \end{cases}$$

$$\mu (Z) = \begin{cases} 0 & ; & x \leq -0,25 \text{ atau } x \geq 0,25 \\ (x - (-0,25)) / (0 - (-0,25)) & ; & -0,25 \leq x \leq 0 \\ (0 - x) / (0,25 - 0) & ; & 0 \leq x \leq 0,25 \end{cases}$$

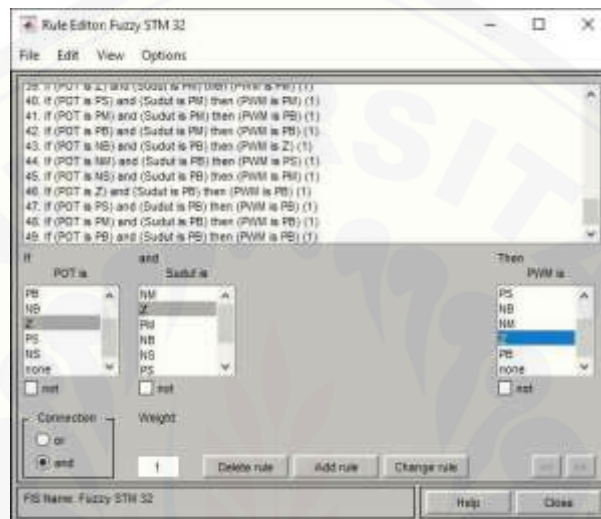
$$\mu (Ps) = \begin{cases} 0 & ; & x \leq 0 \text{ atau } x \geq 2 \\ (x - 0) / (1 - 0) & ; & 0 \leq x \leq 1 \\ (1 - x) / (2 - 1) & ; & 1 \leq x \leq 2 \end{cases}$$

$$\mu (Pm) = \begin{cases} 0 & ; & x \leq 1,5 \text{ atau } x \geq 3,5 \\ (x - 1,5) / (2,5 - 1,5) & ; & 1,5 \leq x \leq 2,5 \\ (2,5 - x) / (3,5 - 2,5) & ; & 2,5 \leq x \leq 3,5 \end{cases}$$

$$\mu (Pb) = \begin{cases} 0 & ; & x \leq 3 \text{ atau } x \geq 6 \\ (x - 3) / (4 - 3) & ; & 3 \leq x \leq 4 \\ 1 & ; & 4 \leq x \leq 5 \\ (6 - x) / (6 - 5) & ; & x \geq 6 \end{cases}$$

b. *Inference Fuzzy*

Pada tahapan ini dilakukan proses pemetaan nilai derajat keanggotaan dari masing-masing variabel *input* yaitu POT dan sudut ke tabel yang berisi basis aturan IF-THEN yang berupa variabel linguistik. Sehingga dapat diambil keputusan terhadap nilai *input* berdasarkan aturan-aturan *rule-base* yang telah dibuat.



Gambar 3.10 *Rule editor* pada Matlab

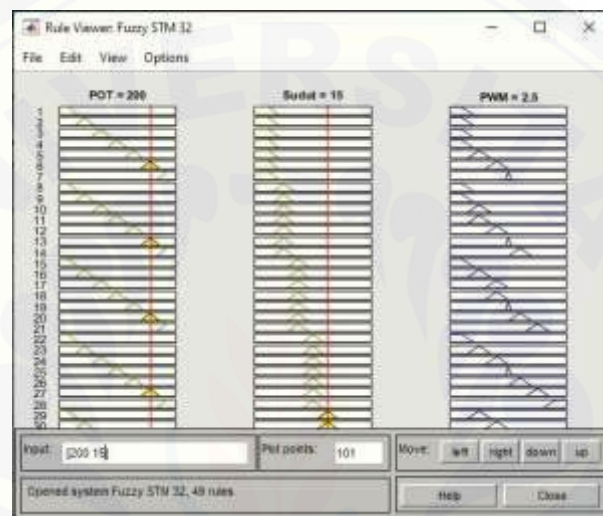
Perancangan *rule editor* digunakan sebagai masukan dengan perancangan 49 *rule-base* yang telah dibuat. Operasi himpunan *fuzzy* dengan logika AND digunakan pada *rule editor* untuk tipe Mamdani. Pada Tabel 3.5 menunjukkan *rule-base* yang digunakan pada penelitian.

Tabel 3.5 *Rule-base* sistem *fuzzy*

POT Sudut	Nb	Nm	Ns	Z	Ps	Pm	Pb
Nb	Nb	Nb	Nb	Nb	Nm	Ns	Z
Nm	Nb	Nb	Nm	Nm	Ns	Z	Ps
Ns	Nb	Nm	Ns	Ns	Z	Ps	Pm
Z	Nb	Nm	Ns	Z	Ps	Pm	Pb
Ps	Nm	Ns	Z	Ps	Ps	Pm	Pb
Pm	Ns	Z	Ps	Pm	Pm	Pb	Pb
Pb	Z	Ps	Pm	Pb	Pb	Pb	Pb

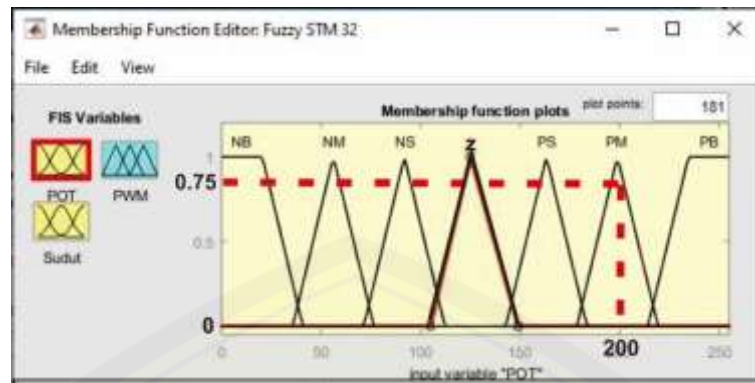
c. *Defuzzification*

Pada tahapan ini dilakukan proses mengubah nilai linguistik hasil inferensi *fuzzy* ke nilai numerik yang dikirim ke *output*. Variabel *output* dari masing-masing aturan yang diberikan bersifat linier dengan variabel *input*. Sehingga pada tahap ini akan menghasilkan satu *output* yang berupa penambahan atau pengurangan PWM setiap detiknya.



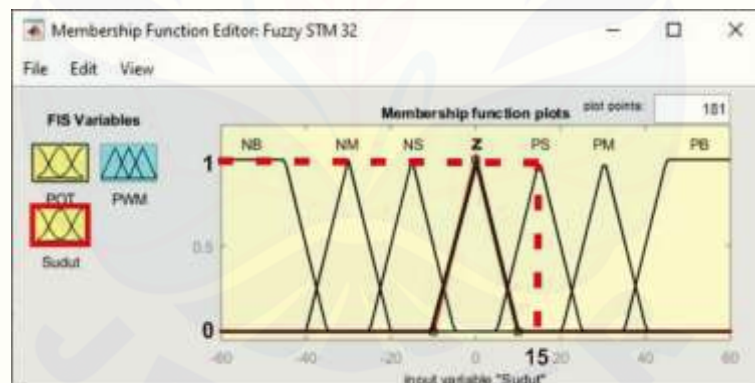
Gambar 3.11 *Rule viewer* pada Matlab

Gambar 3.11 menunjukkan variabel *input* yang berupa POT dan sudut serta variabel *output* berupa PWM. Dari *rule-base* yang telah dibuat diketahui, jika $POT = 200$ dan $sudut = 15^\circ$ maka $output = 2,5$. Berdasarkan hasil untuk menentukan keluaran pada FLC dengan simulasi *rule viewer* pada matlab, maka akan dibandingkan dengan perhitungan defuzzifikasi menggunakan metode *Center of Area* (COA).



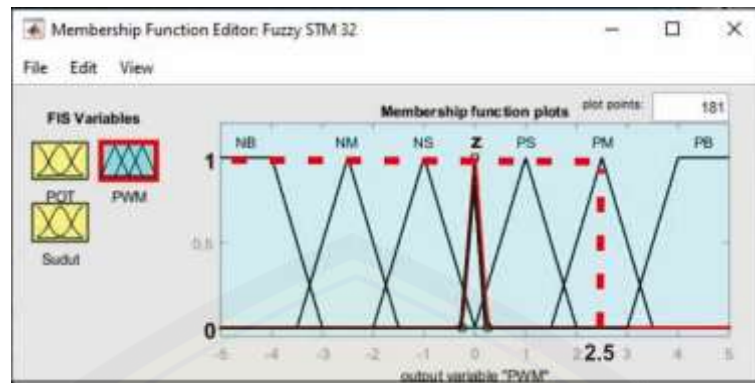
Gambar 3.12 Membership function variabel POT 200

$$\begin{aligned}
 \mu(x)_{PM} &= \frac{c - x}{c - b} \\
 &= \frac{215 - 200}{215 - 195} \\
 &= \frac{15}{20} \\
 &= 0,75
 \end{aligned}$$



Gambar 3.13 Membership function variabel sudut 15°

$$\begin{aligned}
 \mu(y)_{PS} &= \frac{x - a}{b - a} \\
 &= \frac{15 - 5}{15 - 5} \\
 &= \frac{10}{10} \\
 &= 1
 \end{aligned}$$



Gambar 3.14 *Membership function* variabel PWM 2,5

$$\begin{aligned}
 \mu(z) \text{ PM} &= \frac{x - a}{b - a} \\
 &= \frac{2,5 - 1,5}{2,5 - 1,5} \\
 &= \frac{1}{1} \\
 &= 1
 \end{aligned}$$

Selanjutnya membuat *rule-base fuzzy* dengan membandingkan nilai fuzzifikasi pada tiap *input* dengan operator OR. Operator OR berhubungan dengan operasi union atau gabungan pada himpunan. α -predikat sebagai hasil operasi dengan operator OR diperoleh dengan mengambil nilai keanggotaan terbesar antar elemen pada himpunan-himpunan yang bersangkutan.

If POT is PM and Sudut is PS then PWM is PM

$$\begin{aligned}
 \alpha - \text{predikat} &= \mu_{\text{PM}}(x) \cup \mu_{\text{PS}}(y) \\
 &= \text{MAX}(0,75 ; 1) \\
 &= 1
 \end{aligned}$$

Proses terakhir yaitu perhitungan defuzzifikasi dengan metode *Center of Area* (COA) sebagai berikut :

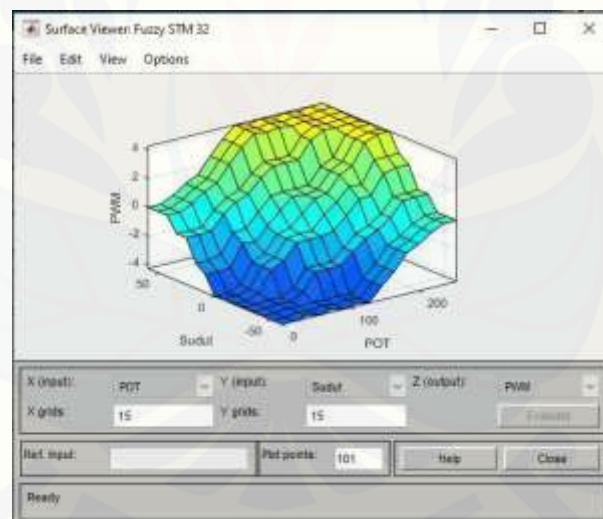
$$\begin{aligned}
 \frac{z1 - a}{b - a} &= \alpha \\
 \frac{z1 - 1,5}{2,5 - 1,5} &= 1 \\
 z1 - 1,5 &= 1 \times 1
 \end{aligned}$$

$$z1 = 1 + 1,5$$

$$z1 = 2,5$$

$$\begin{aligned} \text{Defuzzifikasi (COA)} &= \frac{\sum_{i=1}^1 \alpha \times z1}{\sum_{i=1}^1 \alpha} \\ &= \frac{1 \times 2,5}{1} \\ &= 2,5 \end{aligned}$$

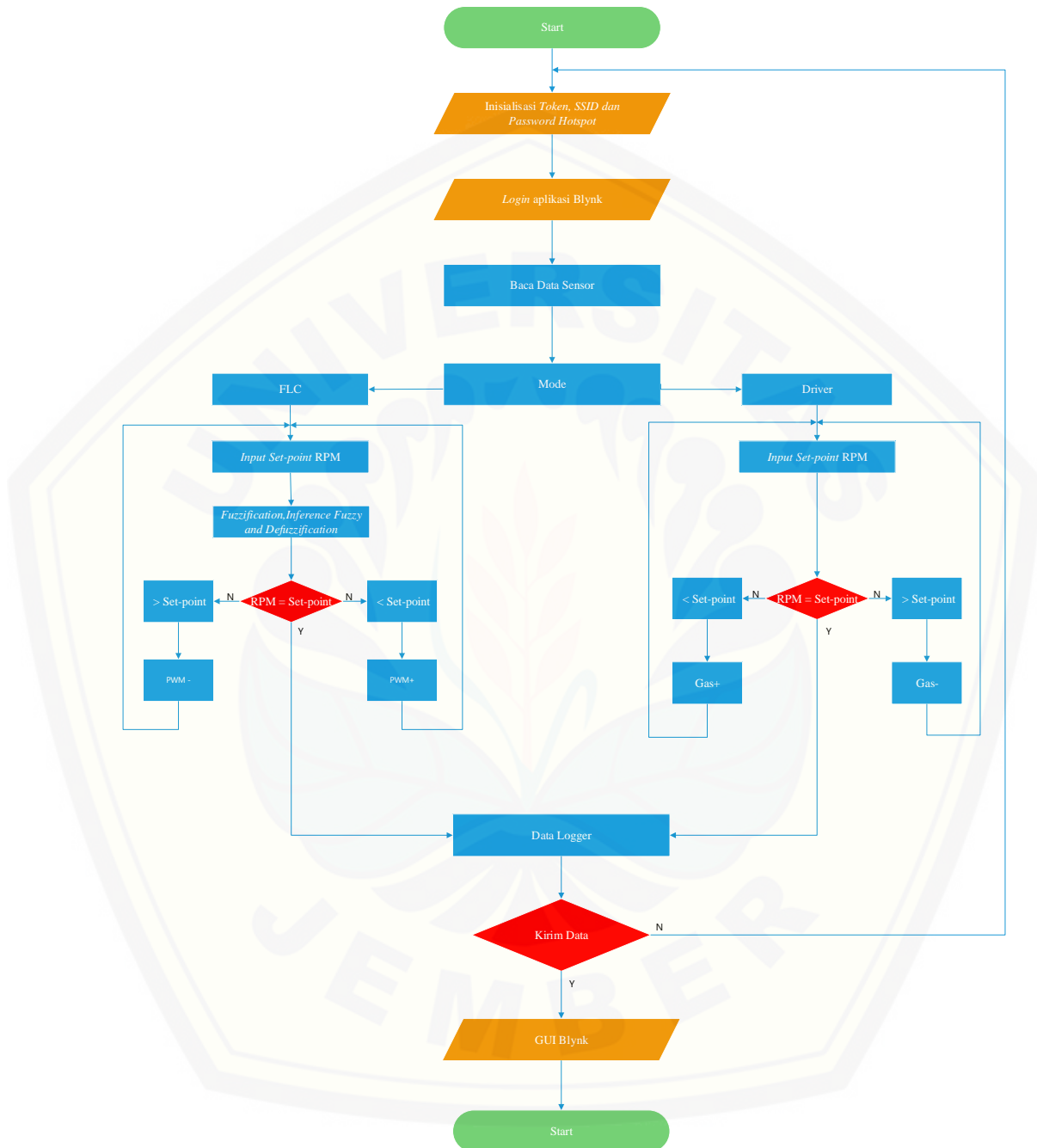
Berdasarkan perbandingan hasil defuzzifikasi pada simulasi Matlab dengan perhitungan defuzzifikasi, bahwa hasilnya sama. Pada simulasi Matlab diperoleh *output* PWM sebesar 2,5, sedangkan hasil perhitungan defuzzifikasi sebesar 2,5. Sehingga hasil defuzzifikasi tersebut akan digunakan sebagai *input driver* MOSFET kontrol BLDC



Gambar 3.15 *Surface viewer* pada matlab

Gambar 3.15 menunjukkan *rule-base* yang telah dibuat dalam bentuk permukaan kendali (*surface viewer*) yang berbentuk linier. Berdasarkan *surface viewer* dapat diketahui respon FLC yang telah dibuat. Semakin linier *surface viewer* maka semakin baik respon kontrolnya.

3.4.5 Flowchart Keseluruhan Sistem



Gambar 3.16 Flowchart keseluruhan sistem

Tahapan sistem kerja alat digunakan sebagai kerangka untuk pengambilan data penelitian. Sistem kerja alat digunakan untuk *monitoring* tegangan, arus, daya,

energi, sudut, pedal gas, PWM, kecepatan, jarak tempuh dan kontrol kecepatan mobil listrik saat kondisi jalan tanjakan dan turunan. Sistem *monitoring* menggunakan *Internet of Things* (IoT) dengan aplikasi Blynk untuk menampilkan data. Sistem kontrol kecepatan terdiri dari *mode driver* dan *mode fuzzy logic controller*. Pada pengujian ini dengan *set-point* sebesar 200 RPM dengan sudut kemiringan jalan sebesar $18,61^\circ$. *Mode driver* yaitu *driver* mengendalikan kecepatan mobil listrik secara langsung dengan menambahkan atau mengurangi pedal gas untuk mengendalikan kecepatan sesuai *set-point*. *Mode FLC* yaitu kecepatan mobil listrik dikendalikan secara otomatis dengan menambahkan atau menurunkan PWM yang masuk ke kontrol BLDC menggunakan algoritma *fuzzy logic controller*. Penambahan atau pengurangan PWM dikendalikan sistem FLC berdasarkan *rule-base*. Variabel *input* berupa nilai ADC potensiometer dan sensor sudut kemudian melalui proses fuzzifikasi, inferensi *fuzzy*, dan defuzzifikasi. Data *output* defuzzifikasi berupa nilai PWM yang digunakan untuk mengatur persentase lebar pulsa *high* terhadap periode pada rangkaian *driver* MOSFET. Apabila kecepatan sesuai dengan *set-point*, maka data dari Arduino DUE disimpan ke data logger dan dikirim ke GUI Blynk. Data logger digunakan untuk menyimpan data tegangan, arus, daya, energi, sudut, pedal gas, PWM, kecepatan, dan jarak tempuh mobil listrik secara *real time*. Proses selanjutnya data dari Arduino DUE dikirim ke GUI Blynk melalui NodeMCU ESP8266. Apabila data telah diterima GUI Blynk, maka proses telah berhasil dilakukan. Apabila pada GUI Blynk belum menerima data, maka proses akan kembali ke proses awal.

3.5 Spesifikasi Motor BLDC

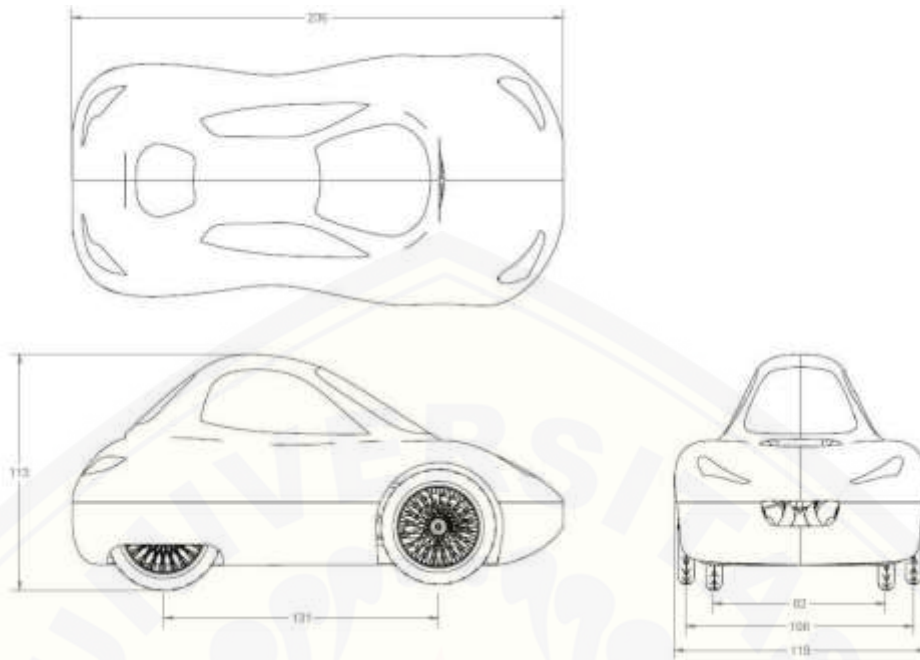
Pada penelitian ini digunakan motor *Brushless Direct Current* sebagai penggerak utama mobil listrik yang dipasang secara *inwheel* di roda belakang mobil listrik. Penggunaan motor *Brushless Direct Current* dalam penelitian ini dikarenakan motor *Brushless Direct Current* memiliki nilai efisiensi yang lebih tinggi dari motor *Direct Current* lainnya. Hal ini sesuai dengan desain kendaraan yang digunakan dalam penelitian. Pada Tabel 3.6 berikut adalah spesifikasi dari motor BLDC.

Tabel 3.6 Spesifikasi motor BLDC

Item	Keterangan
Tipe	Motor <i>Brushless Direct Current</i>
Tegangan	36 V
Daya	250 W
<i>Max. Speed</i>	30 Km/Jam
RPM	400 RPM
Torsi	180 Kg.cm
<i>Controller</i>	Eksternal
Berat	6 Kg
Efisiensi	81%

3.6 Spesifikasi Mobil Listrik

Penelitian ini dilakukan pengujian dinamis dengan mengendarai mobil listrik TITEN EV-3 kategori *Urban Concept* dari Tim Mobil Listrik TITEN Universitas Jember. Desain mobil listrik dengan memperhatikan aerodinamis agar mendapat nilai efisiensi kendaraan yang tinggi. Mobil listrik TITEN EV-3 hanya dikendarai oleh seorang pengemudi, yang disesuaikan dengan regulasi Kompetisi Mobil Hemat Energi di Indonesia. Adapun sumber energi yang digunakan yaitu baterai *Lithium Ion* 36V 10Ah untuk menggerakkan motor *Brushless Direct Current* 250W yang dipasang secara *inwheel* atau motor menyatu dengan roda sehingga tidak memerlukan sistem transmisi mekanik. Gambar 3.17 merupakan desain mobil listrik yang digunakan dalam penelitian.



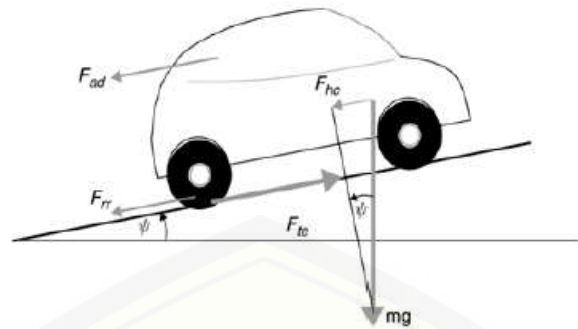
Gambar 3.17 Desain mobil listrik TITEN EV-3

Adapun dimensi mobil listrik yang digunakan pada penelitian ini sebagai berikut :

1. Tinggi mobil listrik adalah 113 cm.
2. Lebar mobil listrik adalah 199 cm.
3. Panjang mobil listrik adalah 236 cm.
4. *Track width* roda depan dengan ukuran 108 cm, sedangkan *track width* roda belakang dengan ukuran 82 cm.
5. Jarak *wheelbase* atau sumbu roda adalah 131 cm.

3.7 Perhitungan *Grade Resistance* Kendaraan

Pada saat kendaraan sedang melaju, maka terdapat gaya hambat salah satunya yang dipengaruhi oleh kemiringan jalan atau *grade resistance*. *Grade resistance* saat menanjak akan melawan gerak kendaraan yang dinyatakan kemiringan positif, sedangkan *grade resistance* saat menurun akan membantu gerak kendaraan yang dinyatakan kemiringan negatif. Penelitian ini mobil listrik melaju pada jalan tanjakan dan turunan dengan sudut maksimal sebesar $18,61^\circ$



Gambar 3.18 *Grade resistance* kendaraan pada kemiringan $18,61^\circ$

Grade resistance kendaraan saat di tanjakan dan turunan pada mobil listrik dapat dihitung dengan rumus matematis sebagai berikut :

$$R_g = \pm m \times g \times \sin \vartheta$$

Keterangan :

R_g = *Grade resistance* (N)

m = Massa Kendaraan (Kg)

g = Percepatan Gravitasi (m/s^2)

ϑ = Sudut Miring ($^\circ$)

Perhitungan *grade resistance* kendaraan saat di tanjakan dan turunan sebagai berikut :

$$\begin{aligned} R_g &= \pm(130 + 70) \times 9,8 \times \sin 18,61^\circ \\ &= \pm 200 \times 9,8 \times 0,319 \\ &= \pm 625,16 \text{ N} \end{aligned}$$

Besarnya *grade resistance* tersebut didistribusikan ke-4 roda sehingga masing-masing roda mendapatkan sebesar 156,29 Newton atau 15,629 kg.

3.8 Desain GUI Blynk



Gambar 3.19 Desain GUI Blynk

Aplikasi Blynk digunakan untuk menampilkan data tegangan, arus, daya, energi, sudut, pedal gas, PWM, kecepatan, dan jarak tempuh mobil listrik. Blynk terhubung dengan NodeMCU ESP8266 dengan inialisasi kode token, SSID dan *Password hotspot*. Kode token didapat saat pertama membuat proyek di Blynk yang dikirim melalui email. SSID dan *Password hotspot* dari *smartphone* yang digunakan agar NodeMCU ESP8266 terhubung dengan jaringan internet. Berikut adalah kode inialisasinya :

1. `char auth[] = "sEvAbgNI_t_bMtBb30d7aeDslL7Ynvjbm"`
2. `char ssid[] = "TITEN JUARA"`
3. `char pass[] = "TITENJuara"`

GUI Blynk terdapat *widget box* yang digunakan untuk menampilkan data. Widget LDC menampilkan data tegangan, arus, daya dan energi. Value display menampilkan pedal gas, kecepatan, PWM, sudut dan jarak tempuh. Superchart

menampilkan grafik tegangan, arus, daya dan kecepatan. Berikut adalah keterangan gambar desain GUI Blynk :

1. Port Koneksi :

Menghubungkan atau memutus sistem *monitoring* dan kontrol antara NodeMCU ESP8266 dengan Blynk.

2. Indikator :

Mengetahui status sistem *monitoring* dan kontrol antara NodeMCU ESP8266 dengan Blynk dengan status *online* atau *offline*.

3. Notifikasi :

Memberitahukan apakah sistem *monitoring* dan kontrol siap digunakan atau belum dengan mengirimkan pesan notifikasi.

4. Data *Monitoring* :

Data yang dikirim dari NodeMCU ESP8266 yang ditampilkan berupa angka dan grafik. Data tersebut meliputi: tegangan, arus, daya, kecepatan, energi, sudut, pegal gas, PWM, dan jarak tempuh mobil listrik.

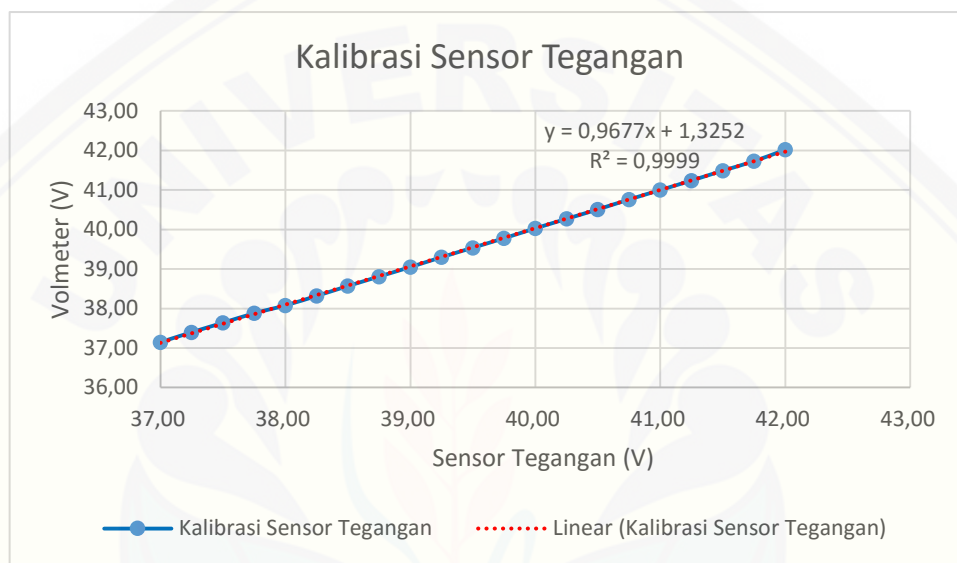
3.9 Kalibrasi Sensor

Kalibrasi Sensor digunakan untuk membandingkan nilai sensor yang digunakan pada penelitian dengan alat ukur yang terkalibrasi agar memperoleh hasil pengukuran sensor mendekati hasil pengukuran alat ukur terkalibrasi. Data hasil pengukuran sensor dan alat ukur dinyatakan dalam grafik fungsi sehingga didapatkan nilai persamaan yang akan digunakan untuk menentukan rumus yang digunakan pada program Arduino DUE.

3.9.1 Sensor Tegangan

Kalibrasi sensor tegangan dilakukan untuk membandingkan nilai pengukuran sensor tegangan yang digunakan pada penelitian dengan *voltmeter* digital. Sensor tegangan yang digunakan dengan konsep pembagi tegangan dua buah resistor yang disusun seri. Resistor yang digunakan sebesar $10\text{ K}\Omega$ dan $1\text{ K}\Omega$ dengan tegangan *input* maksimal 42V , maka diperoleh tegangan *output* sebesar $3,8\text{V}$ yang berupa nilai analog dan dikonversi menjadi nilai digital melalui Arduino

DUE. Perubahan hasil konversi akan dibandingkan dengan hasil pengukuran *voltmeter* digital dengan variasi tegangan untuk mengetahui selisih pengukuran sensor tegangan dengan *voltmeter* digital. Kalibrasi sensor tegangan dimulai dari 37V hingga 42V, karena sumber tegangan dari baterai *lithium ion* yang dipasang *battery management system* (BMS). BMS akan memutus secara otomatis apabila tegangan baterai dibawah 37V atau diatas 42V.



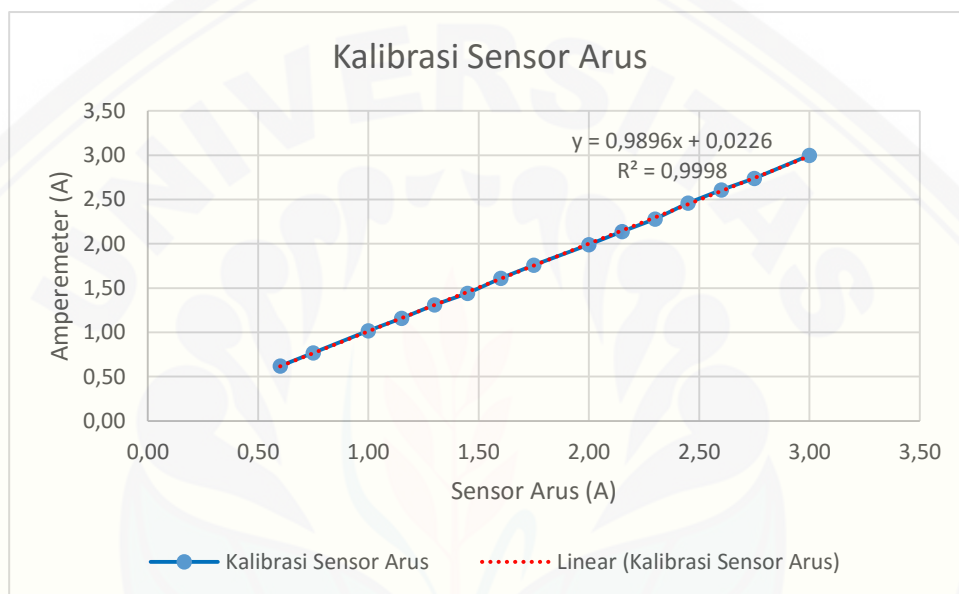
Gambar 3.20 Grafik kalibrasi sensor tegangan

Berdasarkan gambar 3.20 diperoleh nilai persamaan $y = 0,9677x + 1,3252$ dan nilai regresi sebesar 0,9999. Persamaan tersebut digunakan untuk kalibrasi sensor tegangan yang dimasukkan ke dalam program Arduino DUE agar diperoleh hasil pengukuran sensor dengan persentase *error* sekecil mungkin. Nilai regresi menyatakan bahwa tingkat kemiripan atau ketelitian sensor tegangan mendekati 1 sehingga sensor tegangan layak digunakan pada penelitian.

3.9.2 Sensor Arus

Kalibrasi sensor arus dilakukan untuk membandingkan nilai pengukuran sensor arus yang digunakan pada penelitian dengan *amperemeter* digital. Sensor arus yang digunakan pada penelitian dengan jenis ACS712 yang dipasang seri pada

blok sistem *monitoring*. Keluaran dari sensor arus yang berupa nilai analog kemudian dikonversi menjadi nilai digital melalui Arduino DUE. Kalibrasi sensor arus digunakan beban resistor 12,5 Ω 160 Watt dimuali dengan arus 0,60A hingga 3,00A. Perubahan hasil konversi akan dibandingkan dengan hasil pengukuran *amperemeter* digital dengan variasi arus untuk mengetahui selisih pengukuran sensor arus dengan *amperemeter* digital.



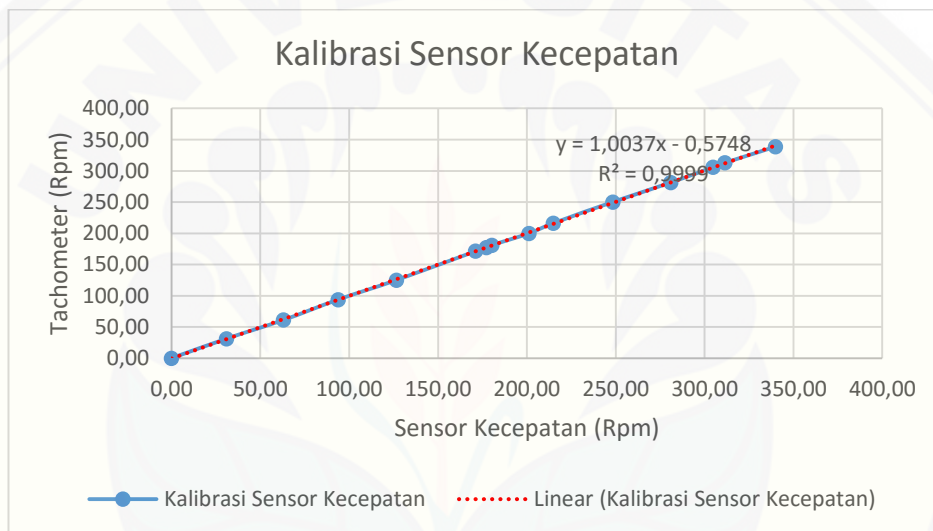
Gambar 3.21 Grafik kalibrasi sensor arus

Berdasarkan gambar 3.21 diperoleh nilai persamaan $y = 0,9896x + 0,0226$ dan nilai regresi sebesar 0,9998. Persamaan tersebut digunakan untuk kalibrasi sensor arus yang dimasukan ke dalam program Arduino DUE agar diperoleh hasil pengukuran sensor dengan persentase *error* sekecil mungkin. Nilai regresi menyatakan bahwa tingkat kemiripan atau ketelitian sensor arus mendekati 1 sehingga sensor arus layak digunakan pada penelitian.

3.9.3 Sensor Kecepatan

Kalibrasi sensor kecepatan dilakukan untuk membandingkan nilai pengukuran sensor kecepatan yang digunakan pada penelitian dengan *tachometer* digital. Sensor kecepatan yang digunakan pada penelitian dengan jenis sensor *hall*

effect. Sensor akan mendeteksi adanya medan magnet sehingga akan mengirimkan sinyal HIGH ke Arduino DUE. Sinyal tersebut dibaca Arduino DUE masuk pin *interrupt* sehingga mengeksekusi secara otomatis dan menghitung waktu setiap sensor mengirim sinyal. Kalibrasi sensor kecepatan dimulai dari kecepatan 0 RPM hingga 340 RPM yang digunakan pada penelitian. Perubahan hasil pengukuran sensor kecepatan akan dibandingkan dengan hasil pengukuran *tachometer* digital dengan variasi kecepatan untuk mengetahui selisih pengukuran sensor kecepatan dengan *tachometer* digital.



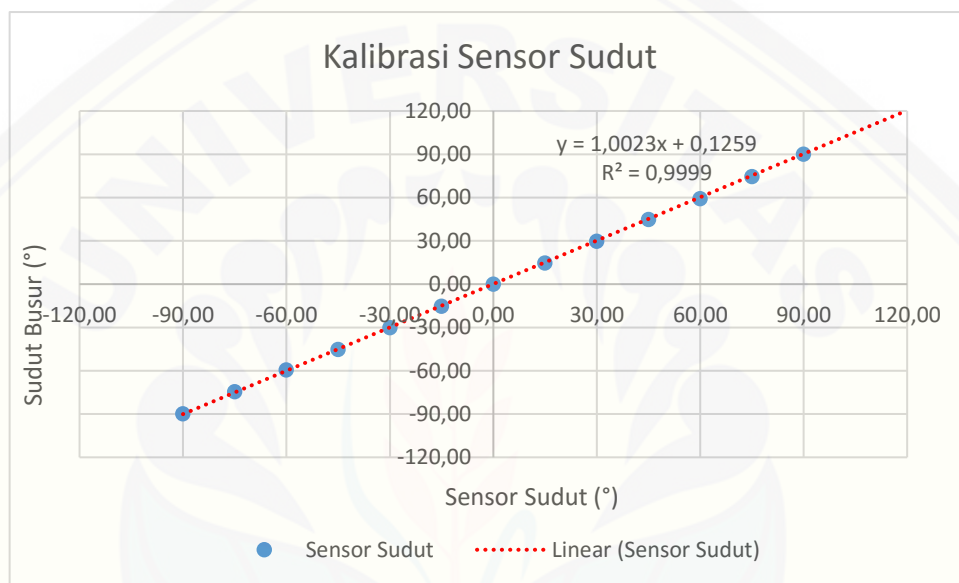
Gambar 3.22 Grafik kalibrasi sensor kecepatan

Berdasarkan gambar 3.22 diperoleh nilai persamaan $y = 1,0037x - 0,5748$ dan nilai regresi sebesar 0,9999. Persamaan tersebut digunakan untuk kalibrasi sensor kecepatan yang dimasukkan ke dalam program Arduino DUE agar diperoleh hasil pengukuran sensor dengan persentase *error* sekecil mungkin. Nilai regresi menyatakan bahwa tingkat kemiripan atau ketelitian sensor kecepatan mendekati 1 sehingga sensor kecepatan layak digunakan pada penelitian.

3.9.4 Sensor Sudut

Kalibrasi sensor sudut dilakukan untuk membandingkan nilai pengukuran sensor sudut yang digunakan pada penelitian dengan penggaris sudut (busur).

Sensor sudut yang digunakan pada penelitian dengan jenis MPU 6050 yang dipasang di penggaris sudut. Kalibrasi sensor sudut dimulai dari sudut -90° hingga 120° . Sudut minus digunakan untuk menunjukkan kondisi turunan dan sudut positif menunjukkan kondisi tanjakan. Perubahan hasil pengukuran sensor sudut akan dibandingkan dengan hasil pengukuran busur dengan variasi sudut untuk mengetahui selisih pengukuran sensor sudut dengan busur.



Gambar 3.23 Grafik kalibrasi sensor sudut

Berdasarkan gambar 3.23 diperoleh nilai persamaan $y = 1,0023x + 0,1259$ dan nilai regresi sebesar 0,9999. Persamaan tersebut digunakan untuk kalibrasi sensor sudut yang dimasukkan ke dalam program Arduino DUE agar diperoleh hasil pengukuran sensor dengan persentase *error* sekecil mungkin. Nilai regresi menyatakan bahwa tingkat kemiripan atau ketelitian sensor kecepatan mendekati 1 sehingga sensor kecepatan layak digunakan pada penelitian.

3.9.5 Jarak Tempuh

Pada pengujian sensor jarak tempuh dilakukan untuk mengetahui jarak yang ditempuh oleh mobil listrik berdasarkan jumlah putaran roda. Sensor yang digunakan pada pengujian ini adalah sensor *hall effect* yang dipasang di dekat roda

dan magnet dipasang di roda. Sensor *hall effect* akan mendeteksi magnet yang berputar mengikuti putaran roda. Setiap 1 kali putaran maka sensor magnet akan mengirimkan 1 kali sinyal HIGH ke Arduino DUE. Roda yang digunakan pada pengujian ini dengan ukuran ban 80/90-16. Adapun makna dari ukuran ban tersebut adalah lebar tapak ban 80 mm, tinggi dinding ban 90 % dari lebar tapak ban atau $90\% \times 80 = 72$ mm dan diameter velg 16 inci.

$$\begin{aligned}\text{Diameter roda} &= (16 \times 25,4) + (80 \times 90\% \times 2) \\ &= 550,4 \text{ mm}\end{aligned}$$

$$\begin{aligned}\text{Keliling roda} &= \pi \times d \\ &= 3,14 \times 550,4 \\ &= 1,728 \text{ m} \\ &= 1,73 \text{ m}\end{aligned}$$

3.10 Pengujian Alat

Pengujian dilakukan secara dinamis atau dengan mengendarai mobil listrik langsung dengan pengujian sistem *monitoring* dan sistem kontrol. Pengujian sistem *monitoring* bertujuan untuk mengetahui kinerja *monitoring*, diantaranya pengujian untuk mengetahui arus terhadap beberapa kondisi jalan. Pengujian sistem kontrol terdiri dari pengujian sistem kontrol FLC dan sistem kontrol tanpa FLC. Pengujian sistem kontrol dilakukan dengan pengujian respon kontrol saat di tanjakan dan turunan serta pengujian penggunaan energi. Pengujian penggunaan energi dilakukan dengan kondisi jalan tanjakan, turunan, belok, dan lurus sejauh 1105,47 meter. Adapun perhitungan penggunaan energi sesuai dengan regulasi lomba Kontes Mobil Hemat Energi (KMHE) dalam satuan km/kwh (KMHE,2019).

$$\text{Penggunaan energi} = \frac{\text{Jarak Tempuh (KM)}}{\text{Energi (KWh)}}$$

BAB 5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil pengujian dan analisa data yang telah dilakukan pada penelitian yang berjudul “Sistem Kontrol Kecepatan dan *Monitoring* Mobil Listrik Berbasis Logika *Fuzzy* dan *Internet of Things*” dapat disimpulkan sebagai berikut :

1. Perancangan sistem kontrol kecepatan dan *monitoring* dapat bekerja dengan baik. Sistem kontrol kecepatan saat di tanjakan dan turunan dengan FLC sedangkan sistem *monitoring* menggunakan IoT yang dikirim ke GUI Blynk.
2. Penggunaan teknologi *Internet of Things* (IoT) pada sistem *monitoring* mobil listrik TITEN EV-3 dapat dilakukan melalui Blynk di mana pun secara *online* dan *real time*.
3. Respon kecepatan saat pengujian tanjakan sistem kontrol tanpa FLC kecepatan rata-rata sebesar 174,90 RPM, sedangkan sistem kontrol FLC sebesar 195,00 RPM dengan *set-point* sebesar 200 RPM. Hal ini dikarenakan, ketika mobil listrik menanjak semakin tinggi, maka *grade resistance* (R_g) yang melawan laju kendaraan menjadi lebih besar. Akan tetapi, sistem kontrol tanpa FLC tidak ada umpan balik PWM yang mengakibatkan kecepatan melambat, sedangkan sistem kontrol FLC terdapat umpan balik penambahan PWM yang mengakibatkan kecepatan bertambah.
4. Respon kecepatan saat pengujian turunan sistem kontrol tanpa FLC kecepatan rata-rata sebesar 280,72 RPM, sedangkan sistem kontrol FLC sebesar 223,59 RPM dengan *set-point* sebesar 200 RPM. Hal ini dikarenakan, ketika mobil listrik menurun diuntungkan oleh *grade resistance* (R_g) yang membantu laju kendaraan. Akan tetapi, sistem kontrol tanpa FLC tidak ada umpan balik PWM yang mengakibatkan kecepatan bertambah, sedangkan sistem kontrol FLC terdapat umpan balik pengurangan PWM yang mengakibatkan kecepatan berkurang.

5. Penggunaan energi pengujian sistem kontrol tanpa FLC sebesar 38,464 km/kwh, sedangkan sistem kontrol FLC sebesar 44,39 km/kwh. Sehingga penggunaan energi sistem kontrol FLC lebih hemat dengan jarak tempuh yang lebih jauh.

5.2 Saran

Berdasarkan penelitian yang telah dilakukan, terdapat beberapa saran yang dapat digunakan untuk pengembangan penelitian selanjutnya yaitu sebagai berikut:

1. Sebaiknya perlu menggunakan mikrokontroler dengan respon yang lebih cepat seperti ARM Cortex-M7.
2. Sebaiknya dilengkapi *Global Positioning System* (GPS) yang dapat mengetahui posisi mobil.
3. Menggunakan sistem kontrol kecepatan saat di tanjakan dan belokan dengan respon kontrol yang lebih baik.

DAFTAR PUSTAKA

- Alazzam, A. dan I. Ahmed. 2019. A speed control system using *Fuzzy logic*. *ITT 2018 - Information Technology Trends: Emerging Technologies for Artificial Intelligence*. 7:12–17.
- Ambildhok, S. P. dan N. B. Hulle. 2017. Development of inertial navigation system based on accelerometer and gyroscope. 7(5):12149–12151.
- Amirullah, M., H. Kusuma, dan T. Tasripan. 2019. Sistem peringatan dini menggunakan deteksi kemiringan kepala pada pengemudi kendaraan bermotor yang mengantuk. *Jurnal Teknik ITS*. 7(2)
- APRESCO, R. A. D. 2017. Perbandingan unjuk kerja notor brushless direct current dan brushed dc pada nogogeni urban konsep. 0:1–71.
- Baatarbileg, A. dan G. M. Lee. 2019. Recent status of electric vehicle charging infrastructure in jeju island. *2019 IEEE Transportation Electrification Conference and Expo, Asia-Pacific (ITEC Asia-Pacific)*. 1–7.
- Bhoi, D. K., P. Sahoo, dan D. S. Patel. 2013. Predictive algorithm to detect uphill or downhill road ahead of vehicle and simulation analysis of impact on fuel economy and drivability. 4(1):1–10.
- Chaudhari, P. S., S. L. Patil, S. K. Pandey, dan S. Sinha. 2017. Performance analysis of bldc motor on sinusoidal and square wave supply. *IEEE International Conference on Power Electronics, Drives and Energy Systems, PEDES 2016*. 2016-Janua:1–6.
- Cladella, F. G. 2017. Perbaikan faktor daya pada pengaturan kecepatan motor bldc menggunakan bridgeless isolated - cuk converter. 92.
- Conker, Ç. 2019. A novel *Fuzzy logic* based safe operation oriented control technique for driving hho dry cell systems based on pwm duty cycle. *International Journal of Hydrogen Energy*. 44(20):9718–9725.
- Dharmawan, A. 2009. Dengan metode pwm sinusoidal menggunakan atmega16
- Ghodake, S., P. Ghadage, N. Patil, dan P. A. Jadhav. 2019. Remotely monitoring health of the solar power system using arduino. 1871–1874.
- Giay, Y. dan T. Adiono. 2019. Design and implementation iot smartplug for electricity usage. *2019 International Symposium on Electronics and Smart Devices (ISESD)*. 1–5.

- Gumelar B, M. A. dan E. Ariyanto. 2017. Implementasi scada untuk monitoring dan controlling serta koordinasi sistem proteksi gardu induk sistem 1,5 breaker pada gardu induk tegangan ekstra tinggi berbasis arduino mega 2560 dengan tampilan hmi. *Gema Teknologi*. 19(3):14.
- Hosseinpour, S., H. Chen, dan H. Tang. 2015. Barriers to the wide adoption of electric vehicles: a literature review based discussion. *Portland International Conference on Management of Engineering and Technology*. 2015-Sept:2329–2336.
- Ikhlas, M. 2015. Rotor position identification for brushless dc motor. *Master of Engineering Science*. 1–103.
- Ismail, Abdul Kharis. 2019. Rancang bangun sistem pengendali kecepatan motor brushless direct current pada kendaraan listrik menggunakan metode Fuzzy inference system
- Jalden, J., X. C. Moreno, dan I. Skog. 2018. USING the arduino due for teaching digital signal processing joakim jalden *, xavier casas moreno kth royal institute of technology , sweden . isaac skog dept . of electrical engineering , linköping . 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 6468–6472.
- Kilickir, H. N., A. Demircali, S. Koroglu, S. Kesler, M. Tumbek, dan A. Kilic. 2017. RF telemetry design for electric vehicle. *6th International Conference on Advanced Technology & Sciences (ICAT'Riga)*. (October):167–170.
- Kushwah, M. dan R. Rani. 2019. Home automation and security system using internet of things. *2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN)*. 235–258.
- Mahmoudi, C., A. Flah, dan L. Sbita. 2014. An overview of electric vehicle concept and power management strategies. *2014 International Conference on Electrical Sciences and Technologies in Maghreb, CISTEM 2014*. (January 2015)
- Masudi, N. 2014. Desain controller motor bldc untuk meningkatkan performa (daya output) sepeda motor listrik. 1–65.
- Olajide, A. O., A. T. Busayo, dan E. T. Olawale. 2017. Analyzing the effects of the different defuzzification methods in the evaluation of javacomponents' customizability for reusability. *International Journal of Advanced Engineering, Management and Science*. 3(9):962–971.
- Padmaja, B. V. S., V. R. Kolluru, dan S. S. Kota. 2019. IoT based implementation of vehicle monitoring and tracking system using node mcu. *International Journal of Innovative Technology and Exploring Engineering*. 8(6):446–450.

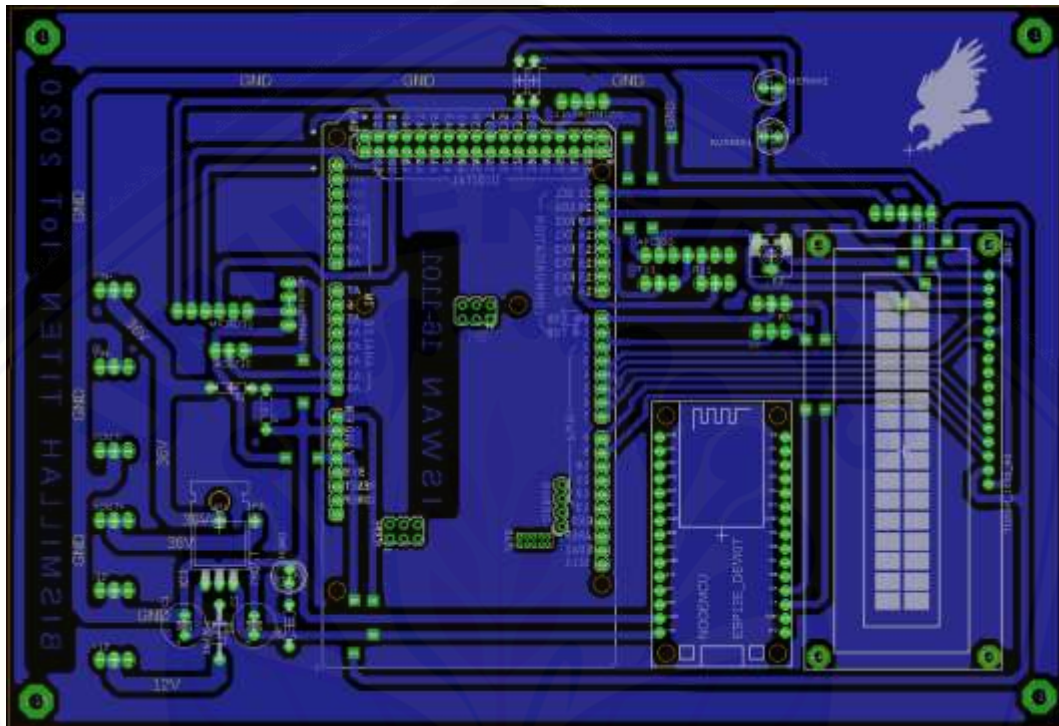
- Peng, Y., X. Zhang, Y. Song, dan D. Liu. 2019. A low cost flexible digital twin platform for spacecraft lithium-ion battery pack degradation assessment. *Conference Record - IEEE Instrumentation and Measurement Technology Conference*. 2019-May(61771157):1–6.
- Pooja Agarwal, P. A. 2013. Brushless dc motor speed control using proportional-integral and Fuzzy controller. *IOSR Journal of Electrical and Electronics Engineering*. 5(5):68–78.
- Poovizhi, M., M. Senthil Kumaran, P. Ragul, L. Irene Priyadarshini, dan R. Logambal. 2017. Investigation of mathematical modelling of brushless dc motor(bldc) drives by using matlab-simulink. *International Conference on Power and Embedded Drive Control, ICPEDC 2017*. 178–183.
- Reddy Avutu, S., S. Paul, V. V. B. A. Prasad, dan J. K. Verma. 2020. Modelling of brushless dc hub motor to control the speed of indigenous powered wheelchair. 091–094.
- Serikul, P., N. Nakpong, dan N. Nakjuatong. 2019. Smart farm monitoring via the blynk iot platform: case study: humidity monitoring and data recording. *International Conference on ICT and Knowledge Engineering*. 2018-Novem:70–75.
- Setiawan, Y., H. Tanudjaja, dan S. Octaviani. 2019. Penggunaan internet of things (iot) untuk pemantauan dan pengendalian sistem hidroponik. *TESLA: Jurnal Teknik Elektro*. 20(2):175.
- Siregar, A. dan I. N. Sutantra. 2018. Analisis kinerja traksi dan redesign transmisi armored personnel carrier komodo 4x4. *Jurnal Teknik ITS*. 7(1):7–11.
- Sujanarko, B., F. Kurniawan, M. Hasan, S. Arifin, dan D. A. Larasati. 2019. Monitoring system and control of university of jember electric vehicle based on radio frequency. 2196–2205.
- Suprianto, G. dan Wirawan. 2018. Implementation of distributed consensus algorithms for wireless sensor network using nodemcu esp8266. *2018 Electrical Power, Electronics, Communications, Controls and Informatics Seminar, EECCIS 2018*. (3):192–196.
- Till Bunsen, P. Cazzola, L. D'Amore, M. Gerner, S. Scheffer, R. Schuitmaker, H. Signollet, J. Tattini, dan J. T. L. Paoli. 2019. Global ev outlook 2019 to electric mobility. *OECD Iea.Org*. 232.
- User, Y. dan N. Tabanli. 2017. Development of induction motor torque control algorithm for electric vehicles on inclined roads. *2017 11th International Conference on Electromechanical and Power Systems, SIELMEN 2017 - Proceedings*. 2017-Janua:472–477.

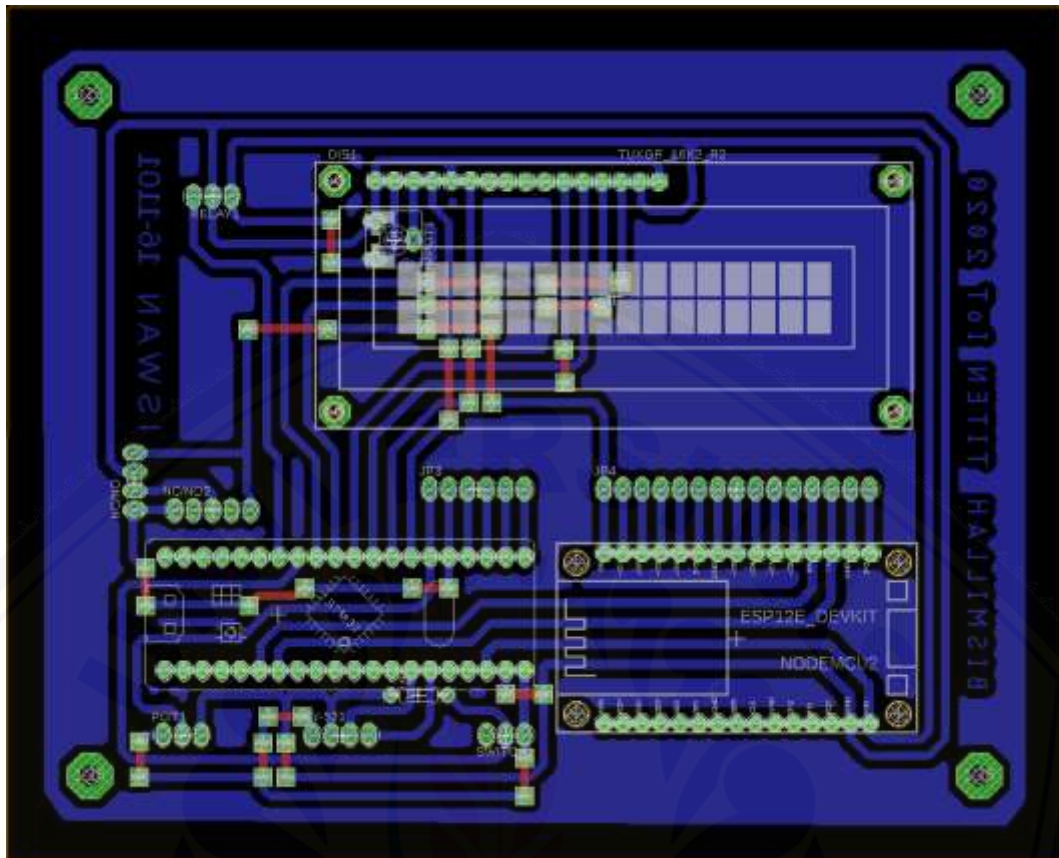
- Vdovic, H., G. S. Member, dan J. Babic. 2019. Automotive software in connected and autonomous electric vehicles : a review. *IEEE Access*. 7:166365–166379.
- Wan, Z., Y. Song, dan Z. Cao. 2019. Environment dynamic monitoring and remote control of greenhouse with esp8266 nodemcu. *Proceedings of 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference, ITNEC 2019*. (It nec):377–382.
- Waworundeng, J. M. S., D. Fernando Tiwow, dan L. M. Tulangi. 2019. Air pressure detection system on motorized vehicle tires based on iot platform. *2019 1st International Conference on Cybernetics and Intelligent System, ICORIS 2019*. 1(August):251–256.
- Wicaksono, A. S. 2016. Perancangan dan implementasi sistem pengaturan kecepatan motor bldc menggunakan kontroler pi berbasis neural-Fuzzy hibrida adaptif. *Jurnal Teknik ITS*. 5(2):68–74.
- Wong, A. 2017. Component selection for an electric 4wd vehicle. 1–10.
- Zeng, B., J. Zhang, L. Chen, dan Y. Wang. 2018. Self-balancing car based on arduino uno r3. *Proceedings of 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference, IAEAC 2018*. (Iaeac):1939–1943.
- Zhang, J., L. Zhang, F. Sun, dan Z. Wang. 2018. An overview on thermal safety issues of lithium-ion batteries for electric vehicle application. *IEEE Access*. 6:23848–23863.
- Zhang, Q. dan G. Li. 2020. Experimental study on a semi-active battery-supercapacitor hybrid energy storage system for electric vehicle application. *IEEE Transactions on Power Electronics*. 35(1):1014–1021.

www.arduino.cc diakses pada 10 Desember 2019

LAMPIRAN

Lampiran 1. Desain PCB Sistem

Gambar 1 Desain PCB sistem *monitoring*



Gambar 2 Desain PCB sistem kontrol

Lampiran 2. Data Pengujian Penggunaan Energi

Lampiran 2. 1 Pengujian penggunaan energi sistem kontrol tanpa FLC

No	Tegangan (V)	Arus (A)	Kecepatan (RPM)	Daya (W)	Energi (Wh)	Jarak (M)
1	39,78	0,20	0,00	7,99	0,02	0,00
2	38,89	6,47	117,87	173,33	0,64	20,76
3	38,22	7,79	157,99	327,39	1,74	41,52
4	37,75	10,90	157,53	448,61	2,21	46,71
5	37,59	10,50	177,07	384,32	4,33	77,85
6	36,81	12,47	177,55	494,54	4,98	119,37
7	37,02	14,00	177,60	459,10	5,12	129,75
8	36,97	13,88	177,69	524,44	5,59	171,27
9	36,76	15,33	177,59	571,07	10,64	186,84
10	36,45	15,37	177,60	561,39	11,14	285,45
11	36,66	14,94	176,57	573,80	11,48	306,21
12	36,45	15,39	177,61	547,63	11,64	311,40

13	36,71	15,36	176,55	569,61	12,65	316,59
14	36,24	15,53	176,56	576,11	13,33	326,97
15	36,55	15,68	176,54	568,35	15,18	337,35
16	36,29	15,37	177,71	562,93	16,03	342,54
17	36,35	14,84	177,59	557,83	16,19	352,92
18	36,71	12,62	177,67	505,92	16,50	368,49
19	37,07	9,14	177,72	446,19	16,78	378,87
20	37,13	10,73	235,48	396,15	17,21	425,58
21	36,76	12,12	235,76	432,03	20,91	591,66
22	36,61	12,15	236,31	433,41	21,17	622,80
23	36,81	11,49	236,40	413,29	21,43	731,79
24	36,97	10,47	236,46	422,85	21,55	752,55
25	37,54	7,03	236,28	337,76	21,77	762,93
26	37,80	5,44	295,60	264,06	21,85	783,69
27	38,01	4,03	295,44	205,77	21,91	788,88
28	38,32	2,39	295,56	153,35	21,95	799,26
29	38,43	2,15	295,58	111,24	22,01	804,45
30	38,37	1,94	295,56	82,58	22,04	814,83
31	38,37	1,47	295,57	74,25	22,06	825,21
32	38,27	3,34	295,66	77,49	22,13	830,40
33	38,32	2,75	295,65	127,90	22,17	861,54
34	38,22	3,87	295,52	105,44	22,20	866,73
35	37,96	5,08	294,72	190,10	22,31	908,25
36	37,85	6,74	295,39	186,31	22,48	918,63
37	37,44	7,85	295,61	300,86	22,65	923,82
38	37,44	7,60	275,97	311,04	22,83	929,01
39	37,59	8,08	275,78	308,74	23,01	944,58
40	37,33	8,73	274,96	322,41	23,19	954,96
41	37,28	8,97	275,97	316,65	23,38	965,34
42	37,23	8,94	275,69	333,92	23,68	970,53
43	36,81	11,82	234,85	421,25	24,60	980,91
44	36,66	11,47	216,75	435,29	24,73	991,29
45	36,92	9,48	196,90	420,47	24,86	1001,67
46	36,92	10,34	197,04	433,31	25,76	1084,71
47	37,59	6,36	196,94	259,38	26,25	1089,90
48	36,50	12,23	196,89	456,88	28,74	1105,47
Rata-rata	37,35	9,39	222,36	358,22	17,44	

Lampiran 2. 2 Pengujian penggunaan energi sistem kontrol FLC

No	Tegangan (V)	Arus (A)	Kecepatan (RPM)	Daya (W)	Energi (Wh)	Jarak (M)
1	39,05	0,20	0,00	8,40	0,02	0,00
2	37,13	9,68	97,89	436,27	0,88	20,76
3	37,39	8,29	138,18	546,78	1,45	41,52
4	37,49	8,49	157,97	309,92	1,54	46,71
5	37,44	9,31	158,10	301,80	2,03	77,85
6	37,18	9,01	177,80	348,47	2,85	119,37
7	38,27	2,66	197,49	289,14	3,04	129,75
8	38,32	2,76	217,14	539,63	3,59	171,27
9	38,74	0,47	217,22	363,74	3,76	186,84
10	36,45	12,13	217,07	435,27	5,97	285,45
11	37,28	7,97	217,20	325,18	6,42	306,21
12	37,33	7,38	217,11	297,08	6,51	311,40
13	37,13	8,00	217,22	275,54	6,59	316,59
14	37,28	7,42	217,15	297,63	6,85	326,97
15	37,39	7,44	217,14	290,47	7,02	337,35
16	37,13	7,98	217,05	278,23	7,11	342,54
17	37,23	8,83	217,27	316,95	7,38	352,92
18	36,87	9,43	217,04	367,24	7,70	368,49
19	36,71	10,30	217,05	370,27	8,02	378,87
20	36,61	11,48	217,28	429,01	9,15	425,58
21	36,40	10,83	217,06	417,58	16,10	591,66
22	36,81	10,09	217,10	482,93	16,63	622,80
23	35,83	15,12	217,14	546,73	19,45	731,79
24	35,88	13,30	217,08	515,07	20,09	752,55
25	37,70	2,91	216,99	498,13	20,38	762,93
26	36,76	8,93	216,00	315,58	20,72	783,69
27	36,76	8,98	217,14	328,27	20,82	788,88
28	37,02	6,71	217,18	320,18	21,01	799,26
29	37,13	6,09	217,06	248,50	21,09	804,45
30	38,06	0,55	217,13	101,08	21,18	814,83
31	37,85	2,07	216,65	31,02	21,20	825,21
32	37,85	1,41	217,19	78,32	21,22	830,40
33	38,06	0,68	276,32	287,85	21,37	861,54
34	38,11	0,58	256,72	25,87	21,38	866,73
35	37,44	4,86	236,86	164,94	21,57	908,25
36	37,49	4,16	236,33	162,49	21,68	918,63
37	38,11	1,04	236,97	155,81	21,72	923,82
38	37,49	4,64	236,96	39,75	21,73	929,01

39	37,33	5,63	237,07	212,78	21,91	944,58
40	37,28	5,62	236,98	200,95	22,03	954,96
41	37,13	6,10	217,13	234,04	22,16	965,34
42	37,07	6,57	217,32	226,64	22,23	970,53
43	37,13	7,15	217,26	255,69	22,45	980,91
44	36,76	8,51	217,14	283,56	22,62	991,29
45	36,92	8,78	217,21	317,86	22,81	1001,67
46	36,14	11,72	217,22	299,62	24,56	1084,71
47	36,35	13,11	217,10	423,55	24,68	1089,90
48	37,49	3,95	217,08	130,78	24,90	1105,47
Rata-rata	37,27	6,86	209,29	294,43	14,53	

Lampiran 3. Listing Program Sistem

Lampiran 3.1 Program Matlab

```

/*
PROGRAM NODEMCU CONTROL
Sensor MPU 6050
Edit by : Hendrik Iswanto NIM 161910201101
15-10-2020
*/

#include "fis_header.h"
// 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 = 1;
// Number of rules to the fuzzy inference system
const int fis_gcR = 49;

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

// Setup routine runs once when you press reset:
void setup()
{
  // initialize the Analog pins for input.
  // Pin mode for Input: ADC
  pinMode(0 , INPUT);
  // Pin mode for Input: Sudut
  pinMode(1 , INPUT);

  // initialize the Analog pins for output.
  // Pin mode for Output: PWM
  pinMode(2 , OUTPUT);
}

```

```

// Loop routine runs over and over again forever:
void loop()
{
  // Read Input: RPM
  g_fisInput[0] = analogRead(0);
  // Read Input: Sudut
  g_fisInput[1] = analogRead(1);

  g_fisOutput[0] = 0;

  fis_evaluate();

  // Set output vlaue: PWM
  analogWrite(2 , g_fisOutput[0]);
}
*****FUZZYFICATION*****

//*****
// 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);
}

// Trapezoidal Member Function
FIS_TYPE fis_trapmf(FIS_TYPE x, FIS_TYPE* p)
{
  FIS_TYPE a = p[0], b = p[1], c = p[2], d = p[3];
  FIS_TYPE t1 = ((x <= c) ? 1 : ((d < x) ? 0 : ((c != d) ? ((d - x) / (d - c)) : 0)));
  FIS_TYPE t2 = ((b <= x) ? 1 : ((x < a) ? 0 : ((a != b) ? ((x - a) / (b - a)) : 0)));
  return (FIS_TYPE) min(t1, t2);
}

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, fis_trapmf
};

// Count of member function for each Input
int fis_gIMFCount[] = { 7, 7 };

// Count of member function for each Output
int fis_gOMFCount[] = { 7 };

// Coefficients for the Input Member Functions
FIS_TYPE fis_gMFI0Coeff1[] = { 35, 55, 75 };
FIS_TYPE fis_gMFI0Coeff2[] = { 175, 195, 215 };
FIS_TYPE fis_gMFI0Coeff3[] = { 210, 230, 250, 250 };
FIS_TYPE fis_gMFI0Coeff4[] = { 0, 0, 20, 40 };
FIS_TYPE fis_gMFI0Coeff5[] = { 105, 125, 145 };
FIS_TYPE fis_gMFI0Coeff6[] = { 140, 160, 180 };
FIS_TYPE fis_gMFI0Coeff7[] = { 70, 90, 110 };
FIS_TYPE* fis_gMFI0Coeff[] = { fis_gMFI0Coeff1, fis_gMFI0Coeff2, fis_gMFI0Coeff3,
fis_gMFI0Coeff4, fis_gMFI0Coeff5, fis_gMFI0Coeff6, fis_gMFI0Coeff7 };
FIS_TYPE fis_gMFI1Coeff1[] = { -40, -30, -20 };
FIS_TYPE fis_gMFI1Coeff2[] = { -10, 0, 10 };
FIS_TYPE fis_gMFI1Coeff3[] = { 20.3174603174603, 30.3174603174603, 40.3174603174603 };
FIS_TYPE fis_gMFI1Coeff4[] = { -60, -60, -45, -35 };
FIS_TYPE fis_gMFI1Coeff5[] = { -25, -15, -5 };
FIS_TYPE fis_gMFI1Coeff6[] = { 5, 15, 25 };
FIS_TYPE fis_gMFI1Coeff7[] = { 35, 45, 60, 60 };

```



```
int fis_gRI16[] = { 7, 5 };
int fis_gRI17[] = { 5, 5 };
int fis_gRI18[] = { 6, 5 };
int fis_gRI19[] = { 2, 5 };
int fis_gRI20[] = { 3, 5 };
int fis_gRI21[] = { 4, 2 };
int fis_gRI22[] = { 1, 2 };
int fis_gRI23[] = { 7, 2 };
int fis_gRI24[] = { 5, 2 };
int fis_gRI25[] = { 6, 2 };
int fis_gRI26[] = { 2, 2 };
int fis_gRI27[] = { 3, 2 };
int fis_gRI28[] = { 4, 6 };
int fis_gRI29[] = { 1, 6 };
int fis_gRI30[] = { 7, 6 };
int fis_gRI31[] = { 5, 6 };
int fis_gRI32[] = { 6, 6 };
int fis_gRI33[] = { 2, 6 };
int fis_gRI34[] = { 3, 6 };
int fis_gRI35[] = { 4, 3 };
int fis_gRI36[] = { 1, 3 };
int fis_gRI37[] = { 7, 3 };
int fis_gRI38[] = { 5, 3 };
int fis_gRI39[] = { 6, 3 };
int fis_gRI40[] = { 2, 3 };
int fis_gRI41[] = { 3, 3 };
int fis_gRI42[] = { 4, 7 };
int fis_gRI43[] = { 1, 7 };
int fis_gRI44[] = { 7, 7 };
int fis_gRI45[] = { 5, 7 };
int fis_gRI46[] = { 6, 7 };
int fis_gRI47[] = { 2, 7 };
int fis_gRI48[] = { 3, 7 };
int* fis_gRI[] = { fis_gRI0, fis_gRI1, fis_gRI2, fis_gRI3, fis_gRI4, fis_gRI5, fis_gRI6, fis_gRI7, fis_gRI8,
fis_gRI9, fis_gRI10, fis_gRI11, fis_gRI12, fis_gRI13, fis_gRI14, fis_gRI15, fis_gRI16, fis_gRI17,
fis_gRI18, fis_gRI19, fis_gRI20, fis_gRI21, fis_gRI22, fis_gRI23, fis_gRI24, fis_gRI25, fis_gRI26,
fis_gRI27, fis_gRI28, fis_gRI29, fis_gRI30, fis_gRI31, fis_gRI32, fis_gRI33, fis_gRI34, fis_gRI35,
fis_gRI36, fis_gRI37, fis_gRI38, fis_gRI39, fis_gRI40, fis_gRI41, fis_gRI42, fis_gRI43, fis_gRI44,
fis_gRI45, fis_gRI46, fis_gRI47, fis_gRI48 };

// Rule Outputs
int fis_gRO0[] = { 4 };
int fis_gRO1[] = { 4 };
int fis_gRO2[] = { 4 };
int fis_gRO3[] = { 4 };
int fis_gRO4[] = { 5 };
int fis_gRO5[] = { 2 };
int fis_gRO6[] = { 6 };
int fis_gRO7[] = { 4 };
int fis_gRO8[] = { 4 };
int fis_gRO9[] = { 5 };
int fis_gRO10[] = { 5 };
```

```
int fis_gRO11[] = { 2 };
int fis_gRO12[] = { 6 };
int fis_gRO13[] = { 3 };
int fis_gRO14[] = { 4 };
int fis_gRO15[] = { 5 };
int fis_gRO16[] = { 2 };
int fis_gRO17[] = { 2 };
int fis_gRO18[] = { 6 };
int fis_gRO19[] = { 3 };
int fis_gRO20[] = { 1 };
int fis_gRO21[] = { 4 };
int fis_gRO22[] = { 5 };
int fis_gRO23[] = { 2 };
int fis_gRO24[] = { 6 };
int fis_gRO25[] = { 3 };
int fis_gRO26[] = { 1 };
int fis_gRO27[] = { 7 };
int fis_gRO28[] = { 5 };
int fis_gRO29[] = { 2 };
int fis_gRO30[] = { 6 };
int fis_gRO31[] = { 3 };
int fis_gRO32[] = { 3 };
int fis_gRO33[] = { 1 };
int fis_gRO34[] = { 7 };
int fis_gRO35[] = { 2 };
int fis_gRO36[] = { 6 };
int fis_gRO37[] = { 3 };
int fis_gRO38[] = { 1 };
int fis_gRO39[] = { 1 };
int fis_gRO40[] = { 7 };
int fis_gRO41[] = { 7 };
int fis_gRO42[] = { 6 };
int fis_gRO43[] = { 3 };
int fis_gRO44[] = { 1 };
int fis_gRO45[] = { 7 };
int fis_gRO46[] = { 7 };
int fis_gRO47[] = { 7 };
int fis_gRO48[] = { 7 };
int* fis_gRO[] = { fis_gRO0, fis_gRO1, fis_gRO2, fis_gRO3, fis_gRO4, fis_gRO5, fis_gRO6, fis_gRO7,
fis_gRO8, fis_gRO9, fis_gRO10, fis_gRO11, fis_gRO12, fis_gRO13, fis_gRO14, fis_gRO15, fis_gRO16,
fis_gRO17, fis_gRO18, fis_gRO19, fis_gRO20, fis_gRO21, fis_gRO22, fis_gRO23, fis_gRO24,
fis_gRO25, fis_gRO26, fis_gRO27, fis_gRO28, fis_gRO29, fis_gRO30, fis_gRO31, fis_gRO32,
fis_gRO33, fis_gRO34, fis_gRO35, fis_gRO36, fis_gRO37, fis_gRO38, fis_gRO39, fis_gRO40,
fis_gRO41, fis_gRO42, fis_gRO43, fis_gRO44, fis_gRO45, fis_gRO46, fis_gRO47, fis_gRO48 };

// Input range Min
FIS_TYPE fis_gIMin[] = { 0, -60 };

// Input range Max
FIS_TYPE fis_gIMax[] = { 250, 60 };

// Output range Min
```

```

FIS_TYPE fis_gOMin[] = { -5 };

// Output range Max
FIS_TYPE fis_gOMax[] = { 5 };

//*****
// 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, 0, 0, 0 };
    FIS_TYPE fuzzyInput1[] = { 0, 0, 0, 0, 0, 0 };
    FIS_TYPE* fuzzyInput[fis_gcI] = { fuzzyInput0, fuzzyInput1 };
    FIS_TYPE fuzzyOutput0[] = { 0, 0, 0, 0, 0, 0 };
    FIS_TYPE* fuzzyOutput[fis_gcO] = { fuzzyOutput0 };
    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_gMFCoeff[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);
    }
}
}
}

```

Lampiran 3.2 Program Arduino

```

/*
PROGRAM NODEMCU CONTROL
Sensor MPU 6050
Edit by : Hendrik Iswanto NIM 161910201101
15-10-2020
*/

#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <SoftwareSerial.h>
#include <SimpleTimer.h>
#include <SD.h> // Kartu SD
#include <SPI.h> //Komunikasi SPI Kartu SD
#define pinCS D8
File myFile;

char auth[] = "T2llcBmO5eksBo7iijJHFFBrG0aa9VtG";
char ssid[] = "TITEN UNEJ"; // Your WiFi credentials.
char pass[] = "TITENJUARA"; // Set password to "" for open networks.

```

```
SimpleTimer timer;
String myString; // complete message from arduino, which consists of sensors data
char rdata; // received characters
float firstVal; int secondVal, thirdVal; // sensors

void myTimerEvent()
{
  // Please don't send more than 10 values per second
  Blynk.virtualWrite(V2, millis() / 1000);
}

BLYNK_WRITE(V9)
{
  int pinSlider = param.asInt();
  pinSlider = map(pinSlider, 0, 255, 0, 1024);
  analogWrite(5, pinSlider);
}

void setup()
{
  Serial.begin(9600);
  Blynk.begin(auth, ssid, pass);
  pinMode(5, OUTPUT);
  timer.setInterval(1000L, sensorSudut);
  timer.setInterval(1000L, adcPotensiometer);
  timer.setInterval(1000L, addPwm);

  if (SD.begin(pinCS))
  {
    Blynk.notify("SD Card Siap");
  }
  else
  {
    Blynk.notify("SD Card Gagal");
    return;
  }
  Blynk.notify("TITEN UNEJ 2020");
}

void loop()
{
  if (Serial.available() == 0 )
  {
    Blynk.run();
    timer.run();
  }
  if (Serial.available() > 0 )
  {
    rdata = Serial.read();
    myString = myString + rdata;
    Serial.print(rdata);
    if ( rdata == '\n')
```

```
{
  String l = getValue(myString, ',', 0);
  String m = getValue(myString, ',', 1);
  String n = getValue(myString, ',', 2);
  firstVal = l.toFloat();
  secondVal = m.toInt();
  thirdVal = n.toInt();

  myFile = SD.open("STM 32 CONTROL.txt", FILE_WRITE);
  if (myFile)
  {
    myFile.print(firstVal);
    myFile.print(",");
    myFile.print(secondVal); //ASLINE RPM
    myFile.print(",");
    myFile.println(thirdVal);
    myFile.print(",");
    myFile.close();
  }

  myString = "";
  Blynk.virtualWrite(V2, rdata); // VIRTUAL PIN APLIKASI BLYNK
}
}
}

//.....SENDING TO BLYNK.....//
void sensorSudut()
{
  float sdata = firstVal;
  // Please don't send more that 10 values per second.
  Blynk.virtualWrite(V7, sdata);
}
void adcpotensiometer()
{
  int sdata = secondVal;
  // Please don't send more that 10 values per second.
  Blynk.virtualWrite(V8, sdata);
}

void addpwm()
{
  int sdata = thirdVal;
  // Please don't send more that 10 values per second.
  Blynk.virtualWrite(V10, sdata);
}

String getValue(String data, char separator, int index)
{
  int found = 0;
  int strIndex[] = { 0, -1 };
  int maxIndex = data.length() - 1;
```

```

for (int i = 0; i <= maxIndex && found <= index; i++) {
  if (data.charAt(i) == separator || i == maxIndex) {
    found++;
    strIndex[0] = strIndex[1] + 1;
    strIndex[1] = (i == maxIndex) ? i + 1 : i;
  }
}
return found > index ? data.substring(strIndex[0], strIndex[1]) : "";
}

/*
PROGRAM MONITORING ARDUINO DUE
Edit by : Hendrik Iswanto NIM 161910201101
15-10-2020
*/

#include <Wire.h> //komunikasi i2c RTC
#include <SD.h> // Kartu SD
#include <SPI.h> //Komunikasi SPI Kartu SD
#include <LiquidCrystal.h>
#include "RTCLib.h"
LiquidCrystal lcd(4, 5, 6, 7, 8, 9); // RS,E,D4,D5,D6,D7
RTC_DS1307 rtc;

//.....INISIALISASI RPM.....//
volatile byte rpmcount;
float rpm;
double timeold;
float datarpmtotal = 0;

//..... INISIALISASI JARAK TEMPUH.....//
int count = 0;
int datacount = -2;
float jarak;

//.....INISIALISASI SENSOR TEGANGAN DAN ARUS.....//
int mVperAmp = 100; //use 100 for 20A Module and 66 for 30A Module
double Voltage = 0;
double arus = 0;
float dataarustotal;
float tegangan;

//.....INISIALISASI WH DAN POWER METER.....//
float msec = 0;
float last_msec = 0;
float elapsed_msec = 0;
float elapsed_time = 0;
float ampSecs = 0;
float ampHours = 0;
float watts = 0;
float wattSecs = 0;

```

```
float wattHours = 0;

int LedHijau = 31;
int LedMerah = 37;
File myFile;
int pinCS = 53;
String cdata;

void rpm_fun()
{
  rpmcount++;
}

void counter()
{
  count++;
}

void setup ()
{
  TCCR1B = TCCR1B & B11111000 | B00000001; // PWM frequency of 31372.55 Hz
  lcd.begin(16, 2);
  Serial.begin(9600); //Nodemcu
  Serial1.begin(9600); //APC220
  attachInterrupt(1, rpm_fun, CHANGE); // Sensor Kecepatan
  attachInterrupt(0, counter, FALLING); // Counter Jarak
  pinMode(pinCS, OUTPUT);
  pinMode(LedHijau, OUTPUT);
  pinMode(LedMerah, OUTPUT);

  rpmcount = 0;
  rpm = 0;
  timeold = 0;

  if (SD.begin())
  {
    lcd.setCursor(0, 0);
    lcd.print("SD Card Siap");
  }
  else
  {
    lcd.setCursor(0, 0);
    lcd.print("SD Card Gagal");
    return;
  }

  // RTC Initialization
  if (!rtc.begin())
  {
    lcd.setCursor(0, 0);
    lcd.print("RTC Bermasalah");
    while (1);
  }
}
```

```

}

if (!rtc.isrunning())
{
  lcd.setCursor(0, 0);
  lcd.print("RTC Daya No");
}
delay(5000);
}

void loop ()
{
  lcd.clear();
  DateTime now = rtc.now();
  PenggunaanEnergi();

  int datarpm;
  datarpmtotal = 0;
  rpm = 0;

  for (int i = 0; i < 500; i++) {
    datarpm = 30 * 700 / (millis() - timeold) * rpmcount; //RPM
  }

  datarpmtotal += datarpm / 500;
  float datarpmtotalfix = datarpmtotal / 100;

  timeold = millis();
  rpmcount = 0;
  attachInterrupt(1, rpm_fun, CHANGE); //Sensor Kecepatan

  //.....KIRIM DATA.....//

  if (count == 0)
  {
    datacount++;
    jarak = datacount * 1.73; //TINGGAL UJI KELILING BAN 20.76 ASLI
    digitalWrite(LedHijau, LOW);
    digitalWrite(LedMerah, HIGH);

    int sensor1 = analogRead(A0); //Tegangan
    attachInterrupt(0, counter, FALLING); //Counter Jarak
    lcd.setCursor(0, 0);
    lcd.print("SIAP KIRIM..."); // MENYIMPAN

    //.....SENDING TO NODEMCU.....//

    cdata = cdata + tegangan + "," + arus + "," + watts + "," + wattHours + "," + datarpmtotal + ","
+ jarak + ",";
    Serial.println(cdata); //Nodemcu

```

```
//.....SENDING TO APC220.....//

Serial1.println(cdata); //APC220 PAKE GUI VB.NET
cdata = "";
}

else
{
  jarak = 0;
  digitalWrite(LedHijau, HIGH); // LED HIJAU
  digitalWrite(LedMerah, LOW); // LED MERAH
  int sensor1 = analogRead(A0); //Tegangan
  attachInterrupt(0, counter, FALLING); // Counter Jarak

  lcd.setCursor(0, 0);
  lcd.print(tegangan);
  lcd.setCursor(6, 0);
  lcd.print("V");
  lcd.setCursor(8, 0);
  lcd.print(arus);
  lcd.setCursor(15, 0);
  lcd.print("A");
  lcd.setCursor(0, 1);
  lcd.print(datarpmtotal);
  lcd.setCursor(5, 1);
  lcd.print("RPM");
  lcd.setCursor(9, 1);
  lcd.print(datacount * 1.73);
  lcd.setCursor(15, 1);
  lcd.print("M");
}

myFile = SD.open("data.txt", FILE_WRITE);
if (myFile)
{
  myFile.print(now.hour(), DEC);
  myFile.print(":");
  myFile.print(now.minute(), DEC);
  myFile.print(":");
  myFile.print(now.second(), DEC);
  myFile.print(",");
  myFile.print(tegangan);
  myFile.print(",");
  myFile.print(arus);
  myFile.print(",");
  myFile.print(datarpmtotal);
  myFile.print(",");
  myFile.print(watts);
  myFile.print(",");
  myFile.print(wattHours);
  myFile.print(",");
}
```

```
myFile.println(jarak);
myFile.print(",");
myFile.close();
}
delay(1000);

if (count >= 2)
{
count = 0;
}
}

void sensorarus()
{
float dataarus[200];
dataarustotal = 0;
arus = 0;

for (int i = 0; i < 200; i++)
{
dataarus[i] = analogRead(A1);
}
dataarustotal = ((dataarus[0] + dataarus[1]..... + dataarus[199]) / 200);
Voltage = (dataarustotal / 1024.0) * 5000;
arus = ((Voltage - 2500) / mVperAmp);
arus = (1.4212 * arus) + 0.01;
if (arus < 0)
{
arus = 0;
}
}

void PenggunaanEnergi()
{
msec = millis();
elapsed_msec = msec - last_msec; //Calculate how long has past since last call of this function
elapsed_time = elapsed_msec / 1000.0; // 1sec=1000 msec
watts = arus * tegangan; //Watts now
ampSecs = (arus * elapsed_time); //AmpSecs since last measurement
wattSecs = ampSecs * tegangan; //WattSecs since last measurement
ampHours = ampHours + ampSecs / 3600; // 1 hour=3600sec //Total ampHours since program started
wattHours = wattHours + wattSecs / 3600; // 1 hour=3600sec //Total wattHours since program started
last_msec = msec; //Store 'now' for next time
}
```