



**MOBIL LISTRIK BERPENGERAK DUA MOTOR DENGAN
PENGATURAN KECEPATAN MENGGUNAKAN SENSOR *GYROSCOPE***

SKRIPSI

Oleh

Kholid Mudzakir Hasan

NIM. 161910201074

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



**MOBIL LISTRIK BERPENGERAK DUA MOTOR DENGAN
PENGATURAN KECEPATAN MENGGUNAKAN SENSOR *GYROSCOPE***

SKRIPSI

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

Oleh

Khold Mudzakir Hasan

NIM. 161910201074

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

PERSEMBAHAN

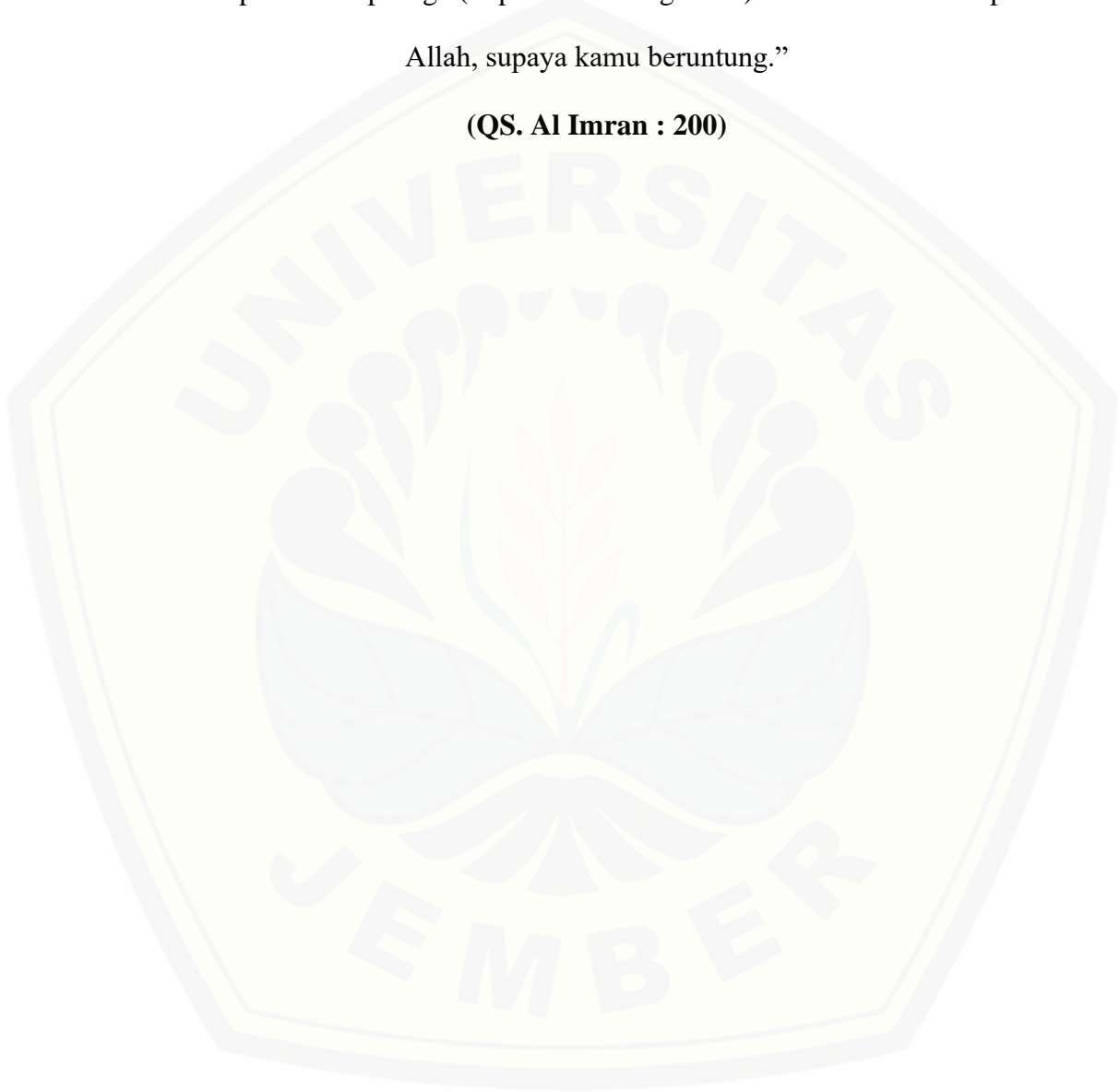
Alhamdulillah puji syukur kehadiran Allah SWT yang telah melimpahkan rahmat dan hidayahNya sehingga saya dapat menyelesaikan penelitian ini. Penulis sangat bersyukur karena telah mendapatkan banyak semangat, bantuan dan doa dari berbagai pihak dalam kelancaran penelitian ini. Akhirnya, skripsi ini dipersembahkan untuk:

1. Kedua Orang tua yang sangat saya sayangi, hormati dan saya banggakan, Bapak Bambang Purwanto dan Ibunda Sopiya. Terimakasih atas segala doa restu, kasih sayang, semangat, dan pengorbanan yang tidak dapat ternilai oleh suatu apapun;
2. Adik-adikku tercinta Syifa Safira dan Misbachussolih, yang selalu memberikan semangat dan doa;
3. Dosen Pembimbing Utama yaitu Prof. Dr. Ir. Bambang Sujanarko, M.M. dan Dosen Pembimbing Anggota Widya Cahyadi, S.T., M.T. yang telah membimbing dengan sabra dan memberikan saran-saran demi kelancaran penelitian ini;
4. Bapak/Ibu Guru sejak Taman Kanak-kanak hingga Sekolah Menengah Atas, dan Bapak/Ibu Dosen Perguruan Tinggi. Terimakasih telah mendidik dan memberikan ilmu maupun pengalaman yang menjadikan saya untuk senantiasa menjadi lebih baik;
5. Almamater tercinta Fakultas Teknik Universitas Jember yang saya banggakan.

MOTTO

“Hai orang-orang yang beriman bersabarlah kamu dan kuatkanlah kesabaranmu dan tetaplah bersiap siaga (di perbatasan negerimu) dan bertakwalah kepada Allah, supaya kamu beruntung.”

(QS. Al Imran : 200)



PERNYATAAN

Saya yang bertanda tangan dibawah ini:

Nama : Kholid Mudzakir Hasan

NIM : 161910201074

Menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “MOBIL LISTRIK BERPENGERAK DUA MOTOR DENGAN PENGATURAN KECEPATAN MENGGUNAKAN SENSOR *GYROSCOPE*” adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada institusi manapun, dan bukan karya jiplakan. Saya bertanggung jawab penuh atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa adanya tekanan dan paksaan pihak manapun serta bersedia mendapat saksi akademik jika ternyata dikemudian hari pernyataan ini tidak benar.

Jember, 30 September 2020

Yang menyatakan,

Kholid Mudzakir Hasan

NIM 161910201074

SKRIPSI

**MOBIL LISTRIK BERPENGERAK DUA MOTOR DENGAN
PENGATURAN KECEPATAN MENGGUNAKAN SENSOR *GYROSCOPE***

oleh

Kholid Mudzakir Hasan

NIM 161910201074

Pembimbing

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

Dosen Pembimbing Anggota : Widya Cahyadi S.T., M.T.

HALAMAN PENGESAHAN

Skripsi berjudul “MOBIL LISTRIK BERPENGGERAK DUA MOTOR DENGAN PENGATURAN KECEPATAN MENGGUNAKAN SENSOR *GYROSCOPE*”

karya Kholid Mudzakir Hasan telah diuji dan disahkan pada:

Hari, tanggal : Selasa, 29 September 2020

Tempat : Fakultas Teknik Universitas Jember

Tim penguji :

Ketua

Anggota I

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

Widya Cahyadi, S.T., M.T.

NIP. 196312011994021002

NIP. 198511102014041001

Anggota II

Anggota III

Dr. Bambang Sri Kaloko, S.T., M.T.

Dr. Ir. Satryo Budi Utomo, S.T., M.T.

NIP. 197104022003121001

NIP. 198501262008011002

Mengesahkan

Dekan,

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

NIP. 1970082619970210011

RINGKASAN

MOBIL LISTRIK BERPENGGERAK DUA MOTOR DENGAN PENGATURAN KECEPATAN MENGGUNAKAN SENSOR *GYROSCOPE*;

Kholid Mudzakir Hasan; 161910201074; 2020; 107 Halaman; Jurusan Teknik Elektro Fakultas Teknik Universitas Jember.

Perkembangan dalam bidang otomotif tertuju pada kendaraan tanpa menggunakan bahan bakar minyak. Dalam melepaskan ketergantungan pada bahan bakar minyak inovasi yang paling utama yaitu mobil listrik. Kendaraan listrik menggunakan motor listrik sebagai tenaga penggerak dimana berfungsi sebagai pengubah energi listrik dalam baterai menjadi energi mekanik untuk memutar roda kendaraan.

Jenis motor yang paling banyak digunakan pada mobil listrik adalah Motor *Direct Current* dan Motor *Brushless Direct Current*. Pada motor *Brushless Direct Current* terdapat keunggulan jika dibandingkan dengan motor listrik yang lain. Keunggulan dari motor BLDC antara lain yaitu efisiensi yang tinggi, suara halus, torsi besar, mudah dikontrol, dan memiliki umur pakai yang panjang.

Penggunaan dua motor pada mobil membuat pemakaian daya semakin bertambah terutama ketika mobil listrik melalui jalan berbelok, daya yang dihasilkan setiap sisi motor berbeda walaupun menggunakan motor yang sama dan dengan tegangan masukan yang sama juga. Hal tersebut dikarenakan beban yang diperoleh motor kanan dan kiri berbeda ketika mobil belok, dalam mengatasi hal tersebut digunakan pengendalian kecepatan setiap sisi motor agar beban yang diterima dapat diminimalisir. Dalam pengendalian ini digunakan metode *Fuzzy Logic* sebagai pengolahan data masukan dan data keluaran sebagai respon terhadap pengendalian kecepatan.

Pada penelitian ini pengembangan mobil listrik dalam segi efisiensi penggunaan energi yaitu dengan penambahan sensor *gyroscope* untuk mengatur suatu kondisi agar konsumsi energi mobil listrik dapat diminimalisasi sehemat

mungkin. Mobil listrik yang dirancang pada penelitian ini memiliki penggerak dua motor BLDC dengan tipe inwheel atau motor sebagai roda.

Kecepatan kedua motor pada mobil listrik ini dipengaruhi oleh dua faktor yaitu ADC pedal gas dan sudut kemudi dengan pembacaan sudutnya menggunakan sensor *gyroscope*. Kedua input tersebut akan diproses melalui mikrokontroler yang kemudian mengirimkan sinyal PWM ke masing-masing *controller* motor kanan dan motor kiri. Dalam pengujiannya ketika mobil dalam keadaan lurus atau sudut kemudi diantara -15 derajat sampai 15 derajat maka PWM yang masuk ke *controller* motor kanan dan motor kiri sama yang mengakibatkan kecepatan kedua motor sama. Ketika mobil berbelok kanan atau sudut kemudi lebih dari 15 derajat maka PWM untuk motor kanan berkurang dan PWM untuk motor kiri sesuai dengan pedal gas, maka dari itu kecepatan dari motor kanan akan lebih pelan dibandingkan dengan motor kiri. Keadaan tersebut sama akan terjadi ketika kemudi belok kiri atau sudut kurang dari -15 derajat, namun pengurangan PWM terjadi pada motor kiri yang mengakibatkan motor kiri lebih pelan dibandingkan dengan motor kanan.

Dalam pengujian efisiensi atau penggunaan daya dilakukan perbandingan mobil listrik menggunakan sistem kontrol dengan sensor *gyroscope* dan mobil listrik tanpa menggunakan sensor *gyroscope*. Pengujian tersebut menggunakan lintasan yang telah ditentukan yang dianggap sesuai dan cocok untuk pengujian sistem ini. Hasil dari pengujian ketika mobil listrik menggunakan sistem dengan sensor *gyroscope* nilai konsumsi daya lebih hemat dibandingkan dengan mobil tanpa sistem.

PRAKATA

Puji syukur ke hadirat Allah SWT atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi dengan judul, “Mobil Listrik Berpenggerak Dua Motor Dengan Pengaturan Kecepatan Menggunakan Sensor *Gyroscope*”. Skripsi ini disusun guna memenuhi salah satu syarat menuntaskan pendidikan Strata Satu (S1) di Jurusan Teknik Elektro Fakultas Teknik Universitas Jember.

Penyusunan skripsi ini tidak lepas dari dukungan dan bantuan dari berbagai pihak, oleh karena itu penulis menyampaikan terima kasih kepada:

1. Allah SWT, atas rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi ini;
2. Nabi besar Muhammad SAW, yang menjadi suri tauladan yang baik bagi seluruh umat;
3. Kedua Orang tua yang sangat saya sayangi, hormati dan saya banggakan, Bapak Bambang Purwanto dan Ibunda Sopiya. Terimakasih atas segala doa restu, kasih sayang, semangat, dan pengorbanan yang tidak dapat ternilai oleh suatu apapun;
4. Adik-adikku tercinta Syifa Safira dan Misbachussolih, yang selalu memberikan semangat dan doa;
5. Bapak Dr. Triwahju Hardianto, S.T, M.T. selaku Dekan Fakultas Teknik Universitas Jember;
6. Bapak Dr. Bambang Sri Kaloko, S.T., M.T. selaku Ketua Jurusan Teknik Elektro Universitas Jember;
7. Bapak Prof. Dr. Ir. Bambang Sujanarko, M.M. selaku Dosen Pembimbing Utama dan Bapak Widya Cahyadi, S.T., M.T. selaku Dosen Pembimbing Anggota yang telah dengan sabar membimbing dan memberikan arahan dalam pengerjaan skripsi;
8. Bapak Dr. Bambang Sri Kaloko, S.T., M.T. dan Bapak Dr. Satryo Budi Utomo S.T., M.T. selaku dosen penguji yang telah memberikan saran untuk memperbaiki dan menyempurnakan skripsi ini;

9. Guru maupun Dosen yang telah memdidik saya dari taman kanak-kanak sampai perguruan tinggi;
10. Keluarga besar Teknik Elektro 2016 (INDUKTRO), terima kasih telah menjadi keluarga yang luar biasa dan senantiasa menemani dalam berproses mulai awal hingga akhir menjadi mahasiswa;
11. Keluarga besar Tim Mobil Listrik (TITEN) Universitas Jember, terimakasih atas ilmu, pengalaman dan kerja sama;
12. Teman-teman satu angkatan yang di LPM, Anam, Andre, Andri, Hendrik, Thofiq, Agung, Dimas, Tanu, Thatit terimakasih telah menjadi sahabat untuk bercerita, berbagi ilmu dan pengalaman.
13. Teman-teman dari IMAKA di jember yang memberikan pengalaman dan pembelajaran dalam perantauan;
14. Teman-teman kontrakan perumahan mastrip mulai dari blok P1, V20 dan O8 terima kasih telah menjadi keluarga selama di jember;
15. Dan semua pihak yang tidak dapat saya sebutkan satu persatu

Jember, 30 September 2020

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PERSEMBAHAN	iii
HALAMAN MOTTO	iv
HALAMAN PERNYATAAN.....	v
HALAMAN PENGESAHAN.....	vii
RINGKASAN	viii
PRAKATA	x
DAFTAR ISI.....	xii
DAFTAR TABEL	xvi
DAFTAR GAMBAR.....	xvii
BAB 1. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan Penelitian	3
1.4 Manfaat Penelitian	3
1.5 Batasan Masalah.....	4
BAB 2. TINJAUAN PUSTAKA.....	5
2.1 Mobil Listrik	5
2.2 Motor <i>Brushless Direct Current</i>.....	6
2.2.1 Gambaran Umum Motor <i>Brushless Direct Current</i>	6
2.2.2 Sistem Motor <i>Brushless Direct Current</i>	7
2.2.3 Prinsip Kerja Motor <i>Brushless Direct Current (BLDC)</i>	7

2.3	<i>Motor Brushless Direct Current Controller</i>	9
2.3.1	<i>Kontrol Kecepatan (Speed Control)</i>	10
2.3.2	<i>Inverter 3 Fasa</i>	10
2.3.3	<i>Modul PWM (Pulse Width Modulation)</i>	11
2.4	<i>Baterai Lead Acid (Aki)</i>	12
2.4.1	<i>Aki Starter (Starting Battery)</i>	13
2.4.2	<i>Aki Deep Cycle Battery</i>	13
2.4.3	<i>Valve Regulated Lead Acid Battery (VRLA)</i>	14
2.5	<i>Arduino</i>	15
2.6	<i>IDE Arduino</i>	16
2.7	<i>Fuzzy Logic</i>	16
2.7.1	<i>Fuzzyfication</i>	17
2.7.2	<i>Rule Evaluation</i>	17
2.7.3	<i>Difuzzyfication</i>	17
2.8	<i>Sensor MPU 6050 Gyroscope and Accelerometer</i>	18
BAB 3.	<i>METODELOGI</i>	20
3.1	<i>Tempat Penelitian</i>	20
3.2	<i>Waktu Penelitian</i>	20
3.3	<i>Tahapan Penelitian</i>	20
3.4.1	<i>Alur Penelitian</i>	20
3.4.2	<i>Diagram Alur Penelitian</i>	22
3.4	<i>Alat dan Bahan Penelitian</i>	22
3.4.1	<i>Alat yang digunakan dalam penelitian ini adalah:</i>	22
3.4.2	<i>Bahan yang digunakan dalam penelitian ini adalah:</i>	23
3.4.3	<i>Software yang digunakan dalam penelitian ini adalah:</i>	23

3.5	Perancangan Alat	23
3.5.1	Perancangan Instalasi Pada Mobil listrik.....	23
3.5.2	Diagram Kerja.....	25
3.5.2	Perancangan Sistem Kontrol	26
3.5.3	Perancangan Perangkat Keras Sistem	27
3.5.4	Perancangan Data Logger	29
3.6	Sistem Kendali <i>Fuzzy</i>	30
3.6.1	Fungsi Keanggotaan <i>Fuzzy</i>	30
3.6.2	<i>Fuzzy Rule</i> Mobil Listrik.....	32
3.7	Metode Pengujian dan Pengambilan Data.....	33
BAB 4. HASIL DAN PEMBAHASAN		35
4.1	Kalibrasi dan Pengujian Sensor.....	37
4.1.1	Sensor Tegangan	37
4.1.2	Sensor Arus	39
4.1.3	Sensor Kecepatan	41
4.2	Kalibrasi dan Pengujian Sensor <i>Gyroscope</i>	43
4.3	Pengujian Sinyal PWM (<i>Pulse Width Modulation</i>).....	48
4.4	Pengujian Nilai RPM Motor BLDC	50
4.5	Pengujian Statis dengan Sistem Kontrol Pada Mobil Listrik	52
4.6	Pengujian Nilai Penggunaan Energi Pada Mobil Listrik	61
4.6.1	Pengujian Searah Jarum Jam.....	62
4.6.2	Pengujian Berlawanan Arah Jarum Jam	67
BAB 5. PENUTUP.....		73
5.1	Kesimpulan	73
5.2	Saran.....	74

DAFTAR PUSTAKA	75
LAMPIRAN.....	77



DAFTAR TABEL

Tabel 2.1 Arduino Uno R3.....	15
Tabel 3.1 Waktu Pelaksanaan Penelitian.....	20
Tabel 3.2 <i>Fuzzy Rule</i>	32
Tabel 4.1 Kalibrasi Sensor Tegangan.....	38
Tabel 4.2 Kalibrasi Sensor Arus.....	39
Tabel 4.3 Kalibrasi Sensor Kecepatan(RPM).....	42
Tabel 4.4 Data Sampel Kalibrasi Sudut Sumbu X.....	44
Tabel 4.5 Hasil Kalibrasi Sudut Sumbu X.....	45
Tabel 4.6 Pengujian Sensor <i>Gyroscope</i> MPU6050.....	47
Tabel 4.7 Data Pengujian Sinyal <i>Pulse Width Modulation</i>	49
Tabel 4.8 Pengujian Kecepatan (RPM) Motor BLDC.....	50
Tabel 4.9 Pengujian Statis Mobil Listrik Kondisi Kemudi Lurus.....	54
Tabel 4.10 Pengujian Statis Mobil Listrik Kondisi Kemudi Belok Kiri.....	56
Tabel 4.11 Pengujian Statis Mobil Listrik Kondisi Kemudi Belok Kanan.....	59

DAFTAR GAMBAR

Gambar 2.1 Rotor Dan Stator Motor BLDC	6
Gambar 2.2 Motor <i>Brushless Direct Current</i> dalam Rangkaian Ekuivalen	7
Gambar 2.3 Tegangan dalam stator BLDC	8
Gambar 2.4 <i>BLDC Controller (Kelly Controllers, 2012)</i>	10
Gambar 2.5 <i>Inverter 3 Fasa</i>	11
Gambar 2.6 Sinyal PWM dengan variasi <i>duty cycle</i>	12
Gambar 2.7 Jenis-Jenis Aki	13
Gambar 2.8 <i>Starting Battery</i>	13
Gambar 2.9 <i>Deep Cycle Battery</i>	14
Gambar 2.10 <i>Valve-Regulated Lead Acid Battery</i>	14
Gambar 2.11 Arduino Uno R3	15
Gambar 2.12 Tampilan IDE Arduino	16
Gambar 2.13 Modul MPU6500	18
Gambar 2.14 <i>Gyroscope Sensor</i>	19
Gambar 2.15 Accelerometer Sensor	19
Gambar 3.1 <i>Flowchart</i> Penelitian	22
Gambar 3.2 Desain Mobil Listrik	24
Gambar 3.3 Rangkaian Wiring Mobil Listrik	24
Gambar 3.4 Diagram Kerja	25
Gambar 3.5 Diagram Blok Sistem Kontrol	26
Gambar 3.6 <i>Flowchart</i> Sistem	27
Gambar 3.7 Sensor MPU6050 dengan Arduino	28
Gambar 3.8 Pedal Gas Dengan Arduino	28
Gambar 3.9 Rangkaian <i>Data Logger</i>	29
Gambar 3.10 Fungsi Keanggotaan Variabel Pedal Gas	30
Gambar 3.11 Fungsi Keanggotaan Variabel Sudut Kemudi	31
Gambar 3.12 Fungsi Keanggotaan PWM Motor Kiri	31
Gambar 3.13 Fungsi Keanggotaan PWM Motor Kanan	32
Gambar 4.1 Tampilan Mobil Listrik Tampak Samping	35

Gambar 4.2 Baterai Pada Mobil Listrik	35
Gambar 4.3 Dua Motor Pada Mobil Listrik	36
Gambar 4.4 <i>Controller Motor</i> dengan <i>Data Logger</i> Pada Mobil Listrik.....	36
Gambar 4.5 <i>Data Logger</i>	37
Gambar 4.6 Kalibrasi Sensor Tegangan	39
Gambar 4.7 Grafik Kalibrasi Sensor Arus	40
Gambar 4.8 Kalibrasi Sensor Kecepatan	41
Gambar 4.9 Grafik Kalibrasi Sensor Kecepatan	43
Gambar 4.10 Grafik Data Sampel Sudut Sumbu X	45
Gambar 4.11 Sensor <i>Gyroscope</i> Pada Kemudi Mobil	48
Gambar 4.12 Grafik Pengujian Kecepatan Motor BLDC.....	52
Gambar 4.13 Grafik Kecepatan Motor Pada Pengujian Statis Dengan Sistem.....	53
Gambar 4.14 Pengujian Statis Mobil Listrik.....	53
Gambar 4.15 Grafik Kecepatan Motor Kondisi Kemudi Mobil Lurus.....	55
Gambar 4.16 Grafik Daya Motor Kondisi Kemudi Mobil Lurus	56
Gambar 4.17 Grafik Kecepatan Motor Kondisi Kemudi Mobil Belok Kiri	58
Gambar 4.18 Daya Motor Kondisi Kemudi Mobil Belok Kiri	58
Gambar 4.19 Grafik Kecepatan Motor Kondisi Kemudi Mobil Belok Kanan	60
Gambar 4.20 Grafik Daya Motor Kondisi Kemudi Mobil Belok Kanan.....	61
Gambar 4.21 Jalur Lintasan Pengujian Secara Dinamis	62
Gambar 4.22 Grafik Kecepatan Motor Pengujian Searah Jarum Jam (Tanpa Sistem)	63
Gambar 4.23 Grafik Daya Motor Pengujian Searah Jarum Jam (Tanpa Sistem) .	64
Gambar 4.24 Grafik Kecepatan Pengujian Searah Jarum Jam (Dengan Sistem) .	65
Gambar 4.25 Grafik Daya Motor Pengujian Searah Jarum jam (Dengan Sistem)	66
Gambar 4.26 Grafik Daya Motor Pengujian Berlawanan Arah Jarum Jam (Tanpa Sistem)	68
Gambar 4.27 Grafik Pengujian Berlawanan Arah Jarum Jam (Tanpa Sistem).....	69
Gambar 4.28 Grafik Kecepatan Pengujian Berlawanan Arah Jarum Jam (Dengan Sistem)	70
Gambar 4.29 Grafik Daya Motor Pengujian Berlawanan Arah Jarum Jam.....	71

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Kebutuhan energi di Indonesia sebagai pendukung perkembangan transportasi dari tahun ke tahun semakin meningkat hal ini dipengaruhi oleh pertumbuhan jumlah penduduk yang terus meningkat. Langkah-langkah inovatif sangat dibutuhkan untuk menanggulangi pasokan bahan bakar yang diprediksi akan terus menurun, tenaga listrik menjadi salah satu solusi alternatif utamanya (Liun, 2018).

Perkembangan dalam bidang otomotif tertuju pada kendaraan tanpa menggunakan bahan bakar minyak. Dalam melepaskan ketergantungan pada bahan bakar minyak inovasi yang paling utama yaitu mobil listrik (Wicaksono dkk., 2016). Kendaraan listrik menggunakan motor listrik sebagai tenaga penggerak dimana berfungsi sebagai pengubah energi listrik dalam baterai menjadi energi mekanik untuk memutar roda kendaraan (Sukerayasa, 2009).

Jenis motor yang paling banyak digunakan pada mobil listrik adalah Motor *Direct Current* dan Motor *Brushless Direct Current*. Namun pada motor *Direct Current* memiliki rugi-rugi gesekan pada komutator karena menggunakan komutator mekanik yaitu berupa sikat sehingga perlu penggantian sikat secara berkala dan pemeliharaan yang rutin. Pada motor *Brushless Direct Current* terdapat keunggulan jika dibandingkan dengan motor listrik yang lain. Keunggulan dari motor BLDC antara lain yaitu efisiensi yang tinggi, suara halus, torsi besar, mudah dikontrol, dan memiliki umur pakai yang panjang. Motor *Brushless Direct Current* beroperasi tanpa menggunakan sikat sehingga rugi gesekan pada sikat dapat dihilangkan (Wicaksono dkk., 2016).

Dalam mobil listrik menggunakan *controller* motor sebagai pendistribusian tegangan dan arus yang diinginkan atau yang dibutuhkan, parameter yang dapat digunakan dalam mengatur besar tegangan masukan ke motor dapat dilakukan dengan *Pulse Width Modulation*. Prinsip kerja PWM dengan mengatur lebar pulsa high terhadap periode dari suatu sinyal dalam bentuk presentase. Namun agar nilai

Pulse Width Modulation dapat diatur secara otomatis maka digunakan pembangkit *Pulse Width Modulation* secara digital.

Analisis energi listrik merupakan upaya untuk mengoptimalkan kerja peralatan pada kondisi beban penuh sehingga penggunaan energi listrik menjadi lebih efektif, efisien dan rasional tanpa harus mengurangi kinerja produksi dan bertujuan untuk mendapatkan informasi mengenai tingkat konsumsi energi yang digunakan per satuan *output* (produksi) serta mengidentifikasi peluang penghematan energi listrik. Pengendalian kecepatan motor merupakan cara yang paling efisien dalam mengendalikan konsumsi daya, sebab jika kecepatan motor berkurang maka pemakaian daya juga berkurang (Malik dan Hariyanto, 2013).

Penggunaan dua motor pada mobil membuat pemakaian daya semakin bertambah terutama ketika mobil listrik melalui jalan berbelok, daya yang dihasilkan setiap sisi motor berbeda walaupun menggunakan motor yang sama dan dengan tegangan masukan yang sama juga. Hal tersebut dikarenakan beban yang diperoleh motor kanan dan kiri berbeda ketika mobil belok, dalam mengatasi hal tersebut digunakan pengendalian kecepatan setiap sisi motor agar beban yang diterima dapat diminimalisir. Dalam pengendalian ini digunakan metode *Fuzzy Logic* sebagai pengolahan data masukan dan data keluaran sebagai respon terhadap pengendalian kecepatan.

Fuzzy logic dapat didefinisikan sebagai logika pengambilan suatu keputusan dalam memecahkan masalah yang bervariasi dengan sistem yang sulit dimodelkan. Sehingga dalam penelitian ini dapat menghasilkan peningkatan efisiensi mobil listrik dengan mengatur kecepatan motor kanan dan motor kiri, dengan perbandingan mobil listrik menggunakan sistem dan mobil listrik tanpa menggunakan sistem tersebut.

Semakin berkembangnya popularitas mobil listrik, akan menarik minat perusahaan otomotif dunia untuk mengembangkannya. Tidak adanya hak milik terhadap mobil listrik, membuat beberapa perguruan tinggi di Indonesia berlomba untuk mengembangkannya. Sejak 2009 pengembangan mobil listrik di Universitas Jember sudah mulai dilakukan, untuk keberlanjutan mobil listrik masih perlu

dilakukan berbagai riset guna mencapai dengan jalan perancangan sistem kelistrikan yang optimal.

Pada penelitian ini pengembangan mobil listrik dalam segi efisiensi penggunaan energi yaitu dengan penambahan sensor *gyroscope* untuk mengatur suatu kondisi agar konsumsi energi mobil listrik dapat diminimalisasi sehemat mungkin.

1.2 Rumusan Masalah

Dari penguraian latar belakang dalam penelitian ini dapat ditarik rumusan masalah antara lain:

1. Bagaimana sistem pengendalian kecepatan mobil listrik berpengerak dua motor menggunakan sensor *gyroscope*?
2. Bagaimana penggunaan energi mobil listrik dengan sistem pengendali kecepatan menggunakan sensor *gyroscope* dan mobil listrik tanpa sistem pengendali kecepatan menggunakan sensor *gyroscope*?

1.3 Tujuan Penelitian

Pada penelitian ini memiliki tujuan antara lain:

1. Mengetahui sistem pengendalian kecepatan mobil listrik berpengerak dua motor menggunakan sensor *gyroscope*.
2. Mengetahui penggunaan energi mobil listrik dengan sistem pengendali kecepatan menggunakan sensor *gyroscope* dan mobil listrik tanpa sistem pengendali kecepatan menggunakan sensor *gyroscope*.

1.4 Manfaat Penelitian

Tercapainya tujuan dari penelitian ini dapat dihasilkan manfaat yaitu mengetahui pengaruh kecepatan dan keluaran daya yang terdapat pada motor listrik atau motor *Brushless Direct Current* saat terjadi perubahan parameter dari nilai PWM yang dipengaruhi oleh nilai sudut kemudi. Diharapkan dengan sistem

pengendali kecepatan ini dapat menghasilkan mobil listrik yang ekonomis dengan konsumsi energi baterai seminimal mungkin dan efisien.

1.5 Batasan Masalah

Batasan dari penelitian ini antara lain:

1. Tegangan yang digunakan adalah 48V.
2. Tidak membahas secara detail tentang konstruksi motor BLDC.
3. Tidak membahas secara detail tentang *controller* motor *Brushless Direct Current*.
4. Motor BLDC yang digunakan memiliki daya nominal sebesar 800W.
5. Kecepatan sudut kendaraan maksimal adalah 600 RPM.
6. Tidak membahas secara menyeluruh tentang rugi-rugi mekanik pada mobil.
7. Pengujian secara dinamis hanya melewati jalan lurus dan belok.
8. Acuan sudut pada kemudi dalam keadaan lurus yaitu rentang sudut -15 derajat sampai 15 derajat.

BAB 2. TINJAUAN PUSTAKA

2.1 Mobil Listrik

Pada tahun 1832-1839 mobil listrik pertama kali dikenalkan oleh Robert Anderson dari Skotlandia, yang ketika itu masyarakat dunia cenderung mengembangkan Motor Bakar yang menggunakan BBM dikarenakan harga bahan bakar minyak masih relatif murah. Pada tahun 2000 para produsen kendaraan mulai benar-benar tertarik pada kendaraan listrik. Hal tersebut dikarenakan harga bahan bakar minyak yang meningkat tinggi dan juga banyak masyarakat yang telah sadar tentang dampak buruknya. Tahun ke tahun persediaan bahan bakar minyak semakin berkurang yang menjadikan harga BBM semakin melonjak. Pengembangan atau inovasi untuk mengganti bahan bakar fosil atau BBM dilakukan pengembangan dalam bidang energi terbarukan terutama pada transportasi yaitu mobil listrik.

Mobil listrik mempunyai banyak keunggulan yang laten dibandingkan mobil dengan mesin yang memiliki sistem pembakaran dalam. Sumber energi dari mobil listrik menggunakan baterai yang tidak menghasilkan gas buang. Tingkat efisiensi energi pada mobil listrik lebih tinggi dibandingkan dengan kendaraan atau mobil bemesin dengan bahan bakar minyak. Pada mobil dengan mesin bakar presentase yang dikonversi menjadi energi kinetik sebagai penggerak kendaraan hanya sekitar 15% saja dan 85% energi yang dihasilkan berupa panas yang terbuang. Berbeda dengan mobil listrik yang memiliki presentase sebaliknya yaitu 86% energi yang digunakan dapat dikonversi menjadi energi kinetik sebagai penggerak kendaraan.

Walupun kendaraan listrik mempunyai banyak keuntungan yang potensial dibandingkan mobil dengan pembakaran dalam, mobil listrik secara luas masih mengalami hambatan dan kekurangan. Harga mobil listrik yang relatif mahal dibandingkan dengan mobil dengan pembakaran dalam atau bemesin menjadi salah satu kelemahan dalam perkembangannya.

2.2 Motor *Brushless Direct Current*

2.2.1 Gambaran Umum Motor *Brushless Direct Current*

Motor *Brushless Direct Current* termasuk jenis motor sinkron yang berarti antara medan magnet stator dan medan magnet rotor berputar pada frekuensi yang sama. Motor *Brushless Direct Current* memiliki beberapa keuntungan dibandingkan dengan motor *Direct Current* biasa, yaitu sebagai berikut :

- Tingkat efisiensi motor yang tinggi.
- Tidak perlu melakukan perawatan terhadap sikat sehingga motor lebih tahan lama.
- Memiliki torsi awal yang tinggi dan kecepatan yang tinggi.
- Tidak menimbulkan suara yang keras.

Motor *Brushless Direct Current* memiliki dua bagian yaitu stator dan rotor. Stator merupakan bagian motor yang tidak bergerak terdiri dari kumparan tiga fasa, dan rotor bagian motor yang bergerak atau berputar terbuat dari permanen magnet.



Gambar 2.1 Rotor Dan Stator Motor BLDC

Sumber tegangan yang diperlukan untuk memutar suatu motor *Brushless Direct Current* (BLDC) menggunakan tegangan berupa *Alternating Current* 3 fasa. Pada implementasinya *Brushless Direct Current* menggunakan sumber tegangan *Direct Current* yang selanjutnya diubah menjadi *Alternating Current* 3 fasa dengan menggunakan *Inverter* 3 fasa, maka dari itu motor ini masih termasuk motor dengan jenis *Brushless Direct Current*. Tegangan AC 3 fasa pada stator motor BLDC akan menghasilkan suatu medan magnet yang akan menarik magnet pada rotor.

2.2.2 Sistem Motor *Brushless Direct Current*

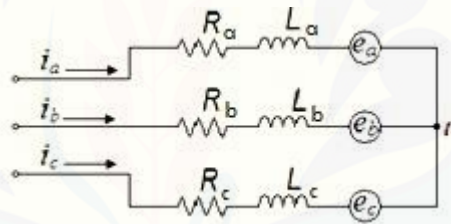
Suatu *Back Electro Magnetic Force* dapat dihasilkan oleh motor *Brushless Direct Current* (BLDC) dengan resistansi stator pada semua lilitan dianggap sama, induktansi diri dan induktansi mutual bernilai konstan. Maka rumus persamaan tegangan tiga fasa sebagai berikut.

$$V_a = R_a I_a + \frac{dia}{dt} L_a + e_a$$

$$V_b = R_b I_b + \frac{dib}{dt} L_b + e_b$$

$$V_c = R_c I_c + \frac{dic}{dt} L_c + e_c$$

Keterangan : $V_{a,b,c}$ = tegangan yang terdapat pada fasa a,b dan c (V)
 $I_{a,b,c}$ = arus yang terdapat pada fasa a,b dan c (A)
 $e_{a,b,c}$ = *back electro magnetic force*



Gambar 2.2 Motor *Brushless Direct Current* dalam Rangkaian Ekuivalen

Memiliki kaitannya dengan medan magnet pada magnet rotor dan interaksi arus pada lilitan stator, dengan torsi elektromagnetik dihasilkan didalam motor BLDC (Pindoriya dkk., 2014).

2.2.3 Prinsip Kerja Motor *Brushless Direct Current* (BLDC)

Motor BLDC dapat bekerja saat stator pada motor diberi tegangan *Alternating Current* tiga fasa, stator tersebut kemudian akan menghasilkan medan magnet (B) dikarenakan stator terbuat dari suatu kumparan.

$$B = \frac{\mu \cdot N \cdot I}{2 \cdot l}$$

Keterangan : N = Banyaknya Lilitan

μ = Permeabilitas Bahan

l = Panjang Lilitan

i = Arus (Ampere)

Nilai Polarisasi setiap kumparan dan medan magnet akan berubah-ubah setiap saat yang mengakibatkan suatu putaran medan magnet.

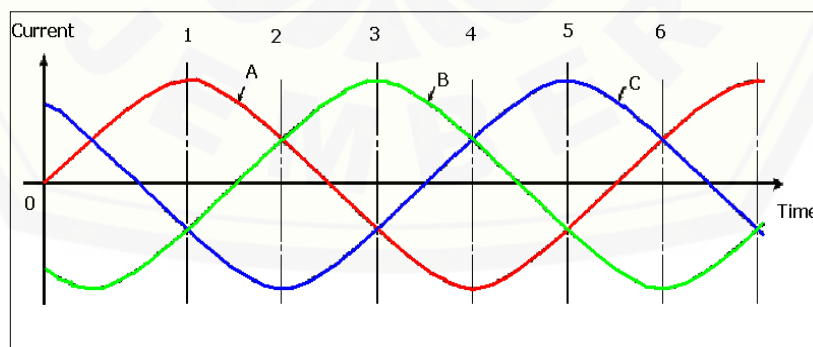
$$N_s = \frac{120 \cdot f}{p}$$

Keterangan, f : Frekuensi tegangan masukan Hz/s

p : jumlah kutub pada stator (*pole*)

Nilai 120 yang terdapat pada rumus diperoleh dari 1 putaran atau 360° per tiga fasa motor (Chritian dan Nugraha, 2013).

Motor *Brushless Direct Current* dapat berputar diakibatkan oleh interaksi antara medan magnet yang dihasilkan oleh stator dengan magnet permanen yang terdapat pada rotor. Dalam menjaga rotasi maka medan magnet yang dihasilkan pada stator harus diputar secara berurutan. (Mpc, 2006). Bentuk sinyal tegangan pada stator motor *Brushless Direct Current* (BLDC) yang terdiri dari sinyal tiga fasa.



Gambar 2.3 Tegangan dalam stator BLDC

Penentuan polaritas pada setiap stator A,B dan C dipengaruhi oleh tegangan masukan setiap fasanya. Ketika salah satu stator diberi tegangan negatif maka akan

menghasilkan medan magnet dengan polaritas negatif, sedangkan kedua stator sisanya diberikan tegangan positif yang kemudian akan menghasilkan polaritas positif. Ketika muncul perbedaan polaritas antara medan magnet kumparan pada stator dengan magnet permanen pada rotor, maka magnet rotor pada sisi positif akan berputar menuju ke medan magnet stator yang berpolaritas negatif. Sedangkan untuk magnet rotor sisi negatif berputar mengikuti stator dengan polaritas positif. Tegangan masukan pada motor berupa *Alternating Current* sinusoidal yang akan mengakibatkan medan magnet untuk stator A,B dan C terus berubah-ubah polaritasnya dengan besar medan magnet tersebut mengikuti perubahan tegangan sinusoidal dari tegangan *Alternating Current*.

Polaritas tegangan pada stator motor dapat berubah secara terus menerus yang diakibatkan oleh tegangan masukan berupa tegangan AC sinusoidal. Dari perubahan polaritas yang terus menerus tersebut kemudian mengakibatkan polaritas medan magnet stator juga berubah-ubah, hal itu akan menjadikan magnet permanen pada rotor bergerak terus menerus mengikuti perubahan polaritas medan magnet stator, atau dapat dikatakan rotor pada motor *Brushless Direct Current* berputar.

2.3 Motor Brushless Direct Current Controller

Dalam pengoperasian motor *Brushless Direct Current* terdapat penunjang utama yaitu *controller* dan *inverter*, fungsi dari *controller* pada motor *Brushless Direct Current* untuk memberikan suatu *trigger* berupa pulsa yang masuk ke stator motor yang kemudian mengatur besar aliran arus sehingga motor dapat berputar dengan akurat.

Terdapat bagian-bagian dalam *Brushless Direct Current Controller* seperti *Speed Control*, *Inverter* dan *PWM Module*.



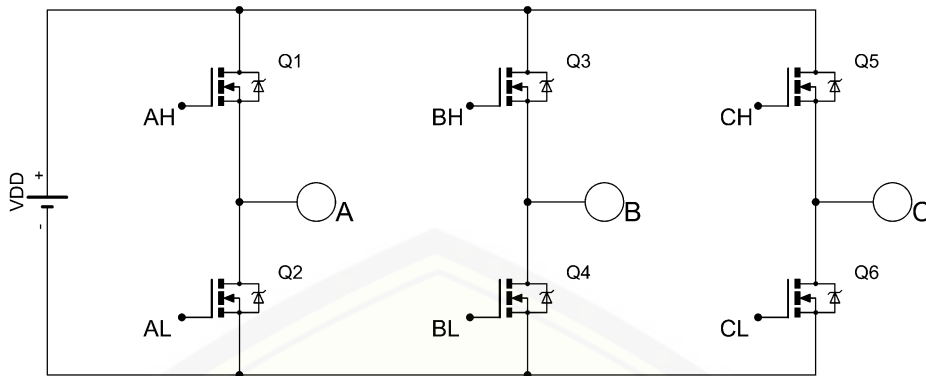
Gambar 2.4 BLDC Controller (Kelly Controllers, 2012)

2.3.1 Kontrol Kecepatan (*Speed Control*)

Controller motor BLDC memiliki unit *speed control* sebagai pengaturan kecepatan motor. Pada umumnya *speed control* yang digunakan pada *controller* motor BLDC menggunakan algoritma kontrol PID. Algoritma PID terdiri dari beberapa mode yaitu Proporsional mode, Integrasi mode, dan Derivatif mode. Dalam menggunakan metode algoritma PID ini memerlukan penentuan parameter pada setiap mode P, I, dan D (Masudi, 2014).

2.3.2 *Inverter* 3 Fasa

Dalam *controller* motor BLDC terdapat *inverter* yang berfungsi mengubah tegangan dan arus listrik yang berasal dari baterai, arus listrik yang diubah adalah arus DC menjadi arus AC. Dalam *inverter* 3 fasa terdapat beberapa komponen yang digunakan untuk switching yaitu dioda yang merupakan komponen utama dalam *inverter*. Pada umumnya jenis diode yang banyak digunakan adalah diode dengan tipe Mosfet, GTO dan IGBT. Pada penyaluran tegangan untuk setiap fasa pada motor diatur oleh diode tersebut yang dihubungkan dengan PWM dimana modul PWM mengirimkan signal kedalam gate pada setiap diode.



Gambar 2.5 Inverter 3 Fasa

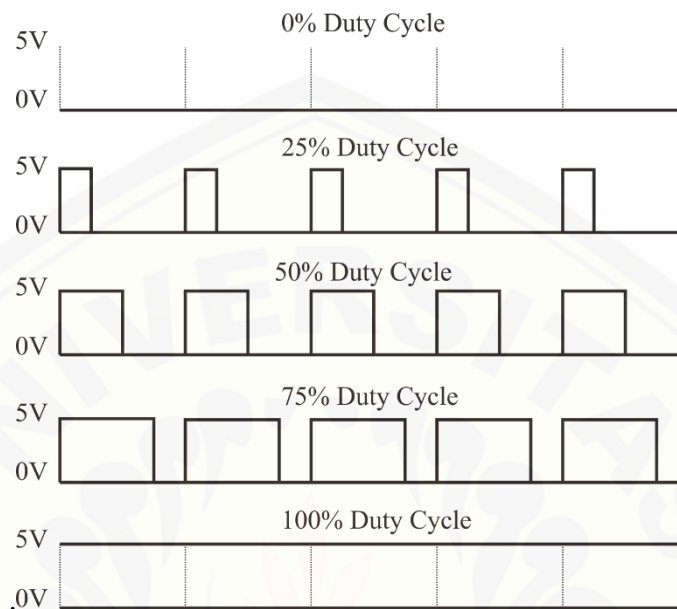
2.3.3 Modul PWM (Pulse Width Modulation)

Pulse Width Modulation memiliki arti modulasi lebar pulsa atau dalam pengertiannya mengatur suatu lebar sinyal dalam bentuk pulsa. PWM banyak digunakan untuk mengontrol suatu tahanan ataupun arus yang mengakibatkan pada pengaturan suatu daya. Peran PWM pada pengendalian putaran atau kecepatan motor terjadi pada pengiriman presentase lebar pulsa terhadap periode. Sinyal yang dihasilkan berupa tegangan periodik dengan bentuk persegi yang selanjutnya masuk ke motor sebagai sumber daya penggerak motor. Cara membangkitkan sinyal PWM dapat digunakan metode analog dan juga digital, untuk metode analog terdapat rangkaian op-amp. Untuk menghasilkan perubahan nilai PWM yang halus metode yang digunakan adalah metode analog. Dalam metode digital jumlah perubahan nilai PWM dipengaruhi oleh resolusi, sebagai contoh jika PWM memiliki resolusi 8 bit maka nilai dapat dijabarkan menjadi 2^8 sama dengan 256 atau nilai perubahan yang dapat dihasilkan 0 sampai 255 dengan rentang *duty cycle* 0% sampai 100%.

$$Duty\ cycle = \frac{T_{on}}{T_{on} + T_{off}} \times 100\%$$

Pada *controller* motor BLDC nilai *Duty cycle* menentukan besar tegangan yang masuk ke motor, jika tegangan sumber 48V dengan *duty cycle* 100% maka tegangan yang masuk ke motor sebesar 48V atau seluruhnya. sinyal tegangan pengatur motor dilewatkan seluruhnya. Jika *duty cycle* sebesar 50% maka tegangan

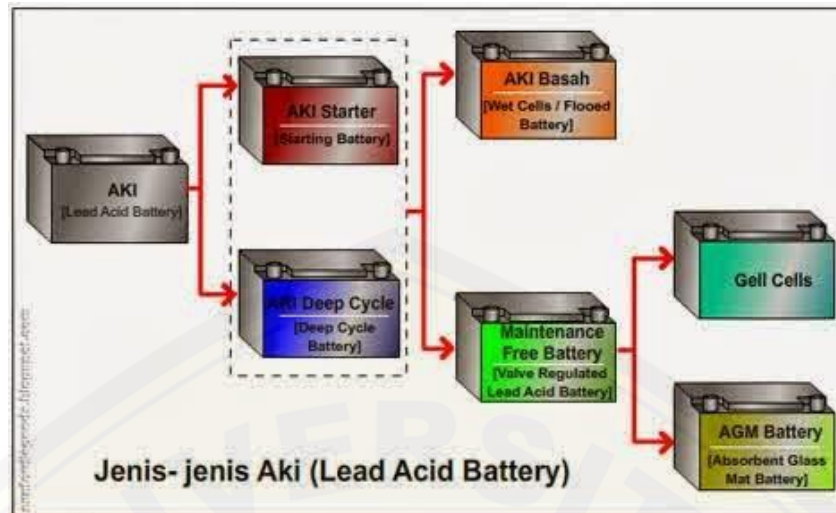
yang masuk ke motor sebesar 24V atau 50% dari tegangan sumber (Nugroho dkk., 2010).



Gambar 2.6 Sinyal PWM dengan variasi *duty cycle*

2.4 Baterai *Lead Acid* (Aki)

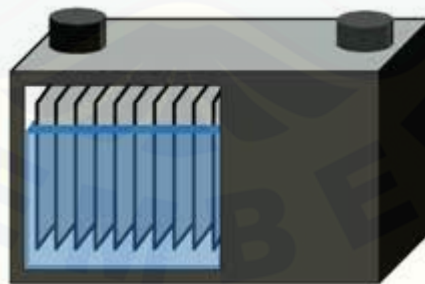
Baterai *Lead Acid* atau sering disebut baterai Aki termasuk baterai dengan bahan utama yaitu bahan kimia. Baterai aki banyak sekali variasinya baik dalam jenis bahan kimianya maupun dari konstruksinya. Dengan variasi yang banyak tersebut menjadikan baterai aki memiliki penggunaan dan perawatan yang berbeda-beda. Penggunaan dan perawatan yang salah dapat mengakibatkan baterai aki tidak tahan lama dan rusak (Fitrianda, 2013). Terdapat dua jenis umum baterai aki yaitu Aki *Starter* dan Aki *Deep Cycle*.



Gambar 2.7 Jenis-Jenis Aki

2.4.1 Aki *Starter (Starting Battery)*

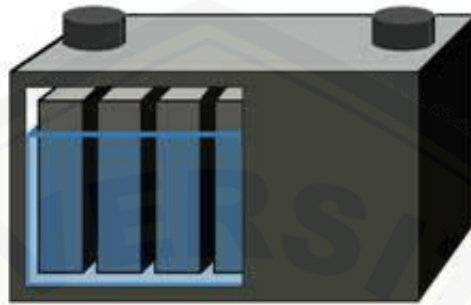
Baterai Aki *Starter* memiliki konstruksi yang terdiri dari banyak plat tipis yang disusun secara paralel yang menjadikan resistansinya rendah, plat ini selain tipis memiliki permukaan yang luas bertujuan untuk dapat mengeluarkan arus listrik yang tinggi. Baterai aki jenis ini dalam waktu singkat dapat menghasilkan arus listrik yang tinggi ketika terdapat beban besar, seperti saat menyalakan mesin pada kendaraan maka baterai ini dapat menghidupkan mesin dalam waktu yang singkat. Baterai akan cepat rusak jika sering digunakan hingga keadaan kosong.

Gambar 2.8 *Starting Battery*

2.4.2 Aki *Deep Cycle Battery*

Baterai aki *Deep Cycle* memiliki konstruksi dengan pelat yang tebal dibandingkan dengan baterai aki *starting*. Jenis baterai aki *Deep Cycle* ini dapat menghasilkan arus listrik yang stabil dalam jangka waktu lama. Dengan pelat yang tebal baterai ini tahan terhadap siklus pengisian baterai yang berulang-ulang,

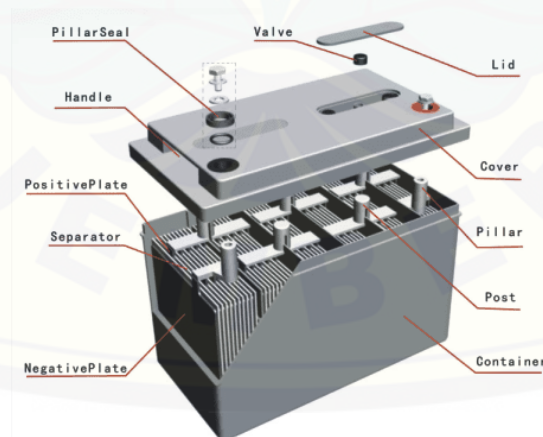
walaupun baterai sering digunakan sampai kapasitas kosong baterai ini lebih awet dibandingkan dengan Aki *Starter*. Pengaplikasian baterai ini sering digunakan pada mobil listrik golf, kursi roda bahkan juga sebagai penyimpanan energi listrik pada proyek pembangkit listrik tenaga surya, angin dan tenaga air.



Gambar 2.9 Deep Cycle Battery

2.4.3 Valve Regulated Lead Acid Battery (VRLA)

Baterai ini sering disebut dengan *Sealed Maintenance Free Battery* atau baterai aki bebas perawatan tanpa terdapat celah untuk isi ulang cairan elektrolit. Baterai aki ini memiliki pelindung fisik kuat yang tertutup dengan rapat bertujuan untuk menjaga cairan elektrolit dalam aki tidak bocor. Bagian luar yang terlihat pada aki ini hanya terminal kutup positif (+) dan terminal kutup negatif (-).



Gambar 2.10 Valve-Regulated Lead Acid Battery

Baterai Aki VRLA memiliki dua jenis yang dibedakan berdasarkan konstruksi dalamnya yaitu jenis *Gel Cells* dan *Absorbent Glass Mat (AGM)*.

2.5 Arduino

Arduino Uno memiliki arsitektur dengan jenis perangkat elektronik open source. Arduino memiliki keunggulan daripada board mikrokontroler yang lain salah satu keunggulannya bahasa yang digunakan dalam program adalah bahasa C. Terdapat 14 pin pada Arduino berupa yang terdiri dari 14 pin I/O dengan pin keluaran PWM berjumlah 6 pin. Dilengkapi port USB yang mempermudah pemrograman mikrokontroler Arduino melalui perangkat lain.



Gambar 2.11 Arduino Uno R3

Terdapat beberapa rincian spesifikasi dari Arduino Uno R3 yang diuraikan dalam tabel 2.1 berikut.

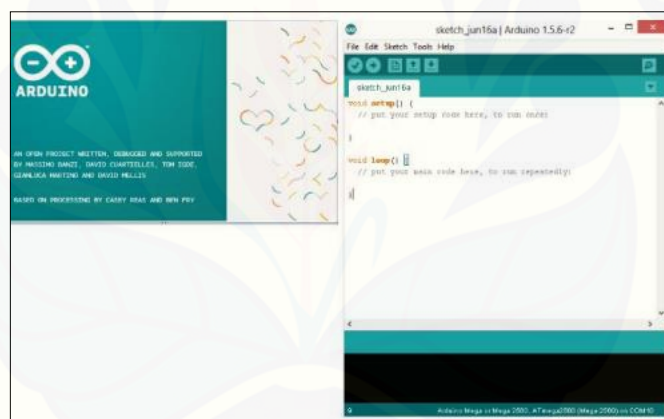
Tabel 2.1 Arduino Uno R3

Mikrokontroler	ATmega328
Tegangan operasi	5V
Tegangan <i>input</i> (disarankan)	7-12V
Digital I/O	14 pin (dimana 6 output PWM)
<i>Input</i> Analog	6 pin
Arus DC per I/O	Pin 40 mA
Arus DC untuk 3.3V	Pin 50 mA
SRAM	2 KB (ATmega328)
EPROM	1 KB (ATmega328)
Kecepatan <i>clock</i>	16 MHz

2.6 IDE Arduino

IDE Arduino merupakan software yang digunakan untuk pemrograman dalam mikrokontroler pada Arduino. Software ini menggunakan bahasa C++ yang sangat mudah karena disajikan dalam bentuk library. IDE Arduino memiliki tiga bagian yaitu editor program, compiler, uploader.

Editor program untuk menuliskan maupun mengedit listing program (sketch) dalam bahasa processing. Compiler bertujuan untuk mengubah listing program dalam bahasa processing menjadi kode berupa biner yang dimengerti oleh mikrokontroler dan uploader yang mengirimkan biner tersebut ke memori mikrokontroler. Secara umum terdapat dua perintah pada Arduino yaitu pertama void setup yang memiliki perintah yang jalankan sekali saja dan kedua void loop yang memiliki perintah yang dijalankan secara berulang-ulang (Ismail, 2019).



Gambar 2.12 Tampilan IDE Arduino

2.7 Fuzzy Logic

Fuzzy Logic Control merupakan kontrol dari proses pengambilan keputusan dengan aturan yang berfungsi untuk memecahkan permasalahan sulit, ambigu atau tidak jelas. Logika *Fuzzy* tidak mengenal logika yang tegas yaitu benar atau salah. *Fuzzy Logic* termasuk kontrol cerdas karena dapat memecahkan masalah dari suatu sistem tanpa bantuan manusia secara real time serta dapat menyelesaikan masalah yang kompleks, stasioner, dan tidak linier.

Perancangan kontrol *fuzzy* harus memperhatikan faktor yang mendasari dari sistem *fuzzy*. Faktor mendasar tersebut adalah penskalaan dari *Input/output* aturan dasar dari pengendali *fuzzy* dan tipe fungsi keanggotaan *fuzzy* yang digunakan. Penskalaan *Input/output* terlebih dahulu ditentukan sebelum menentukan tipe fungsi keanggotaan.

Sampai sekarang logika *fuzzy* masih banyak digunakan dikarenakan memiliki sifat yang fleksibel dan dapat menerima data yang kurang tepat, memiliki konsep yang gampang dimengerti.

2.7.1 *Fuzzyfication*

Fuzifikasi berfungsi untuk memudahkan dalam pengelompokan *Input fuzzy* sesuai dengan himpunannya, agar dapat diketahui nilai masukannya maka pemetaan atau pengelompokan diperlukan suatu model dari fungsi keanggotaannya (Kristiyono dkk., 2015). Terdapat beberapa jenis keanggotaan yang digambarkan seperti gaussian, segitiga, dan trapesium.

2.7.2 *Rule Evaluation*

Rule evaluation atau aturan penilaian merupakan langkah kedua dalam logika *fuzzy*. *Rule evaluation* ini memproses nilai pada *Input* yang telah difuzifikasi dan selanjutnya dimasukkan pada *rule-rule* yang telah ditentukan dalam basis aturan. Dari nilai beberapa *Input* yang telah dimasukkan ke *rule fuzzy* maka akan menghasilkan *output*, jumlah nilai *output* juga telah ditentukan dari basis aturan sebelumnya.

Pada umumnya permodelan dalam aturan penilaian terdapat dua model yaitu model Mamdani dan model Sugeno. Dalam aturan *fuzzy* terdapat inti dari relasi *fuzzy* yang disajikan dalam bentuk IF-THEN.

2.7.3 *Difuzzyfication*

Difuzzifikasi merupakan proses penegasan dari besaran *fuzzy* yang berupa *membership function* dalam himpunan *fuzzy*. Karena dalam aplikasi nyata penentuan nilai yang pasti sangat dibutuhkan saat nilai keluaran *fuzzy* setelah proses *rule* yang berikutnya dimasukkan pada variabel *output*. Dalam penentuan himpunan fungsi keluaran digunakan garis lurus yang ditarik secara vertikal dengan nilai keluaran *fuzzy* digambarkan sebagai derajat dari *output membership function*. Hasil

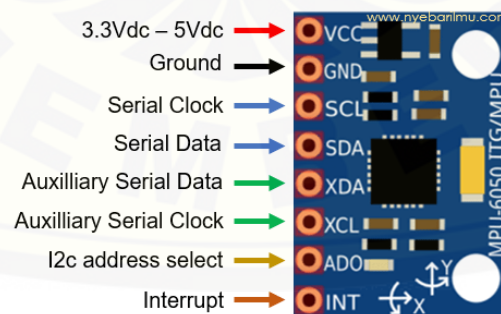
tersebut akan dimasukkan pada rumus berdasarkan permodelannya. Pada model mamdani dikenal beberapa rumus yang digunakan, yaitu *Center Of Gravity* (COG), *Bisector of Area*, *Centroid of Area*, *Mean of Maximum*. Model sugeno mempunyai 2 rumus, yang pertama adalah model Orde Nol dan yang kedua model Orde Satu. Rumus tersebut akan menghasilkan hasil akhir atau crisp.

2.8 Sensor MPU 6050 Gyroscope and Accelerometer

Sensor Gyroscope dan Accelerometer dengan tipe MPU 6050 berasal dari InveSense yang dirangkai dalam satu chip dengan konstruksi *Microelectromechanical System*. Sensor MPU 6050 mampu mencatat nilai chanel untuk x,y, dan z dengan sangat akurat dalam waktu bersamaan dan mampu bekerja dengan begitu cepat dan presisi (InvenSense Inc., 2014).

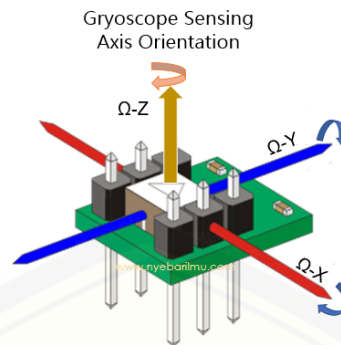
Spesifikasi dari *module* MPU6500 yaitu sebagai berikut :

1. Menggunakan chip IC MPU6050
2. Tegangan operasional di *range* antara 3Vdc – 5Vdc
3. Komunikasi menggunakan I2C (SCL, SDA)
4. Rentang dari Gyroscope sensor sebesar 250 500 1000 dan 2000 / detik
5. Mempunyai 16 bit AD *converter*, data keluaran 16 bit
6. Difungsikan sebagai sensor *accelerometer* dan *gyroscope*
7. Ukuran modul : 2.2cm x 1.7cm



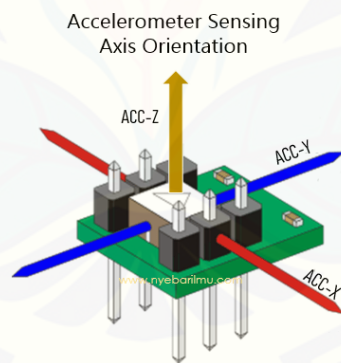
Gambar 2.13 Modul MPU6500

Sensor *gyroscope* memiliki fungsi dalam penentuan besar sudut secara cepat yang bertumpuan pada suatu poros sebagai sumbu dalam rotasinya.



Gambar 2.14 Gyroscope Sensor

MPU6050 untuk bagian sensor accelerometer mampu untuk pengukuran percepatan pada objek yang bergerak secara dinamis maupun statis. Untuk mengukur besar percepatan pada benda atau objek yang bergerak dinamakan pengukuran secara dinamis, sedangkan pada statis pengukuran percepatan dilakukan pada benda yang mengacu pada gravitasi bumi. Dalam pengaplikasiannya sensor accelerometer banyak digunakan pada pengukuran kemiringan sudut yang secara mendasar menggunakan prinsip acceleration.



Gambar 2.15 Accelerometer Sensor

BAB 3. METODELOGI

3.1 Tempat Penelitian

Pelaksanaan penelitian dilakukan di Labolatorium *Renewable Energy*, Universitas Jember dan Workshop Tim Mobil Listrik TITEN.

3.2 Waktu Penelitian

Dalam pelaksanaan penelitian dikerjakan dalam waktu beberapa bulan dengan rincian sebagai berikut:

Tabel 3.1 Waktu Pelaksanaan Penelitian

No	Kegiatan	Bulan								
		1	2	3	4	5	6	7	8	9
1.	Studi Literatur	■	■							
2.	Perancangan Alat		■	■	■	■				
3.	Penelitian dan pengujian				■	■	■	■		
4.	Analisa Data						■	■	■	■
5.	Penulisan Laporan							■	■	■

3.3 Tahapan Penelitian

3.4.1 Alur Penelitian

Dalam proses penelitian tugas akhir ini ada beberapa tahapan yang dilakukan, berikut adalah langkah-langkah dalam penelitian ini yaitu :

a. Studi literatur

Pada tahap ini merupakan tahapan untuk mencari sumber informasi terkait dengan penelitian yang dilakukan, dengan informasi yang didapat dari berbagai sumber untuk menjadi acuan mencapai hasil penelitian ini.

b. Perancangan perangkat keras

Perancangan perangkat keras yang dilakukan adalah merancang mobil listrik dengan system alat sesuai dengan studi literatur yang didapatkan.

c. Perancangan perangkat lunak

Perancangan perangkat lunak yang dilakukan adalah perancangan komponen-komponen yang dibutuhkan pada mobil listrik dan pemrograman dengan *software* sebagai pengolahan parameter yang digunakan.

d. Implementasi keseluruhan sistem

Setelah perancangan perangkat keras dan perangkat lunak tahap selanjutnya adalah menggabungkan keseluruhan sistem. Langkah selanjutnya adalah dilakukan pengujian sistem.

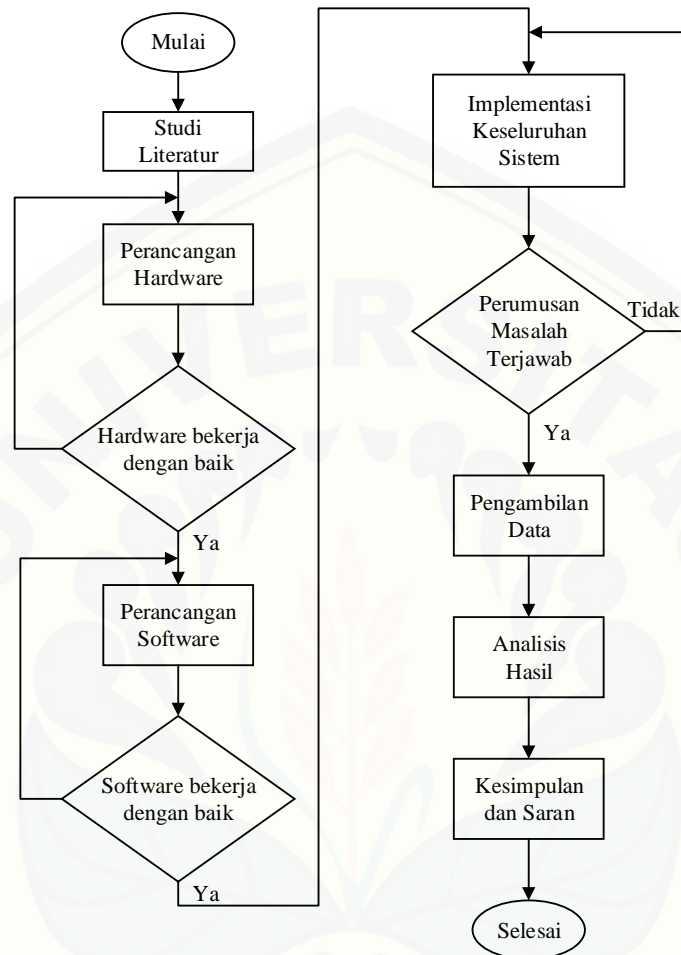
e. Pengambilan data

Pada tahapana ini setelah semua sistem siap untuk diuji dilakukan pengambilan data dengan pengujian yang berbeda-beda untuk mendapatkan data.

f. Analisa data

Setelah tahap pengujian dan pengambilan data dilakukan dan didapatkan data, maka langkah selanjutnya yaitu melakukan analisis terhadap data yang didapat.

3.4.2 Diagram Alur Penelitian



Gambar 3.1 Flowchart Penelitian

3.4 Alat dan Bahan Penelitian

3.4.1 Alat yang digunakan dalam penelitian ini adalah:

- Sensor MPU6050 *Gyroscope* and Accelerometer
- Arduino UNO R3
- Baterai *Lead acid* 48V
- Motor *Brushless Direct Current* 800 Watt
- Kelly *Motor Controller Brushless Direct Current* 2000 Watt
- Pedal Gas / *Throttle*
- Wattourmeter*

- h. *Tachometer*
 - i. Solder
 - j. Penyedot timah
 - k. AVO meter
 - l. Laptop
 - m. *Data Logger*
- 3.4.2 Bahan yang digunakan dalam penelitian ini adalah:
- a. Timah
 - b. Kabel
 - c. Mur
 - d. Baut
- 3.4.3 Software yang digunakan dalam penelitian ini adalah:
- a. Arduino IDE

3.5 Perancangan Alat

3.5.1 Perancangan Instalasi Pada Mobil listrik

Alat ini menggunakan sensor MPU6050 *Gyroscope and Accelerometer* yang ditempatkan bagian kemudi pada mobil listrik sebagai pembacaan sudut kemudi ketika mobil belok. Terdapat pedal gas sebagai pembatas kecepatan mobil, baterai aki 48V sebagai sumber energi, dua *controller* dan dua motor BLDC sebagai penggerak mobil listrik. Kemudian digunakan MCB sebagai pengaman ketika mobil mengalami over *Current*, juga digunakan *watthourmeter* sebagai pembacaan besar tegangan, arus, Wh, dll.



Gambar 3.2 Desain Mobil Listrik



Gambar 3.3 Rangkaian Wiring Mobil Listrik

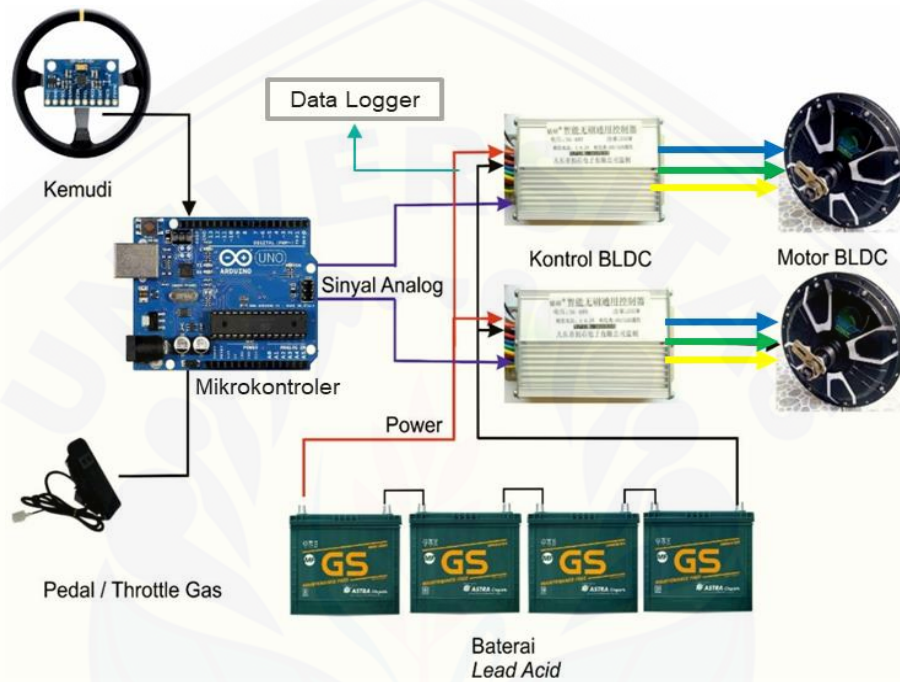
Keterangan

1. *Throttle* / Pedal gas
2. Baterai 48 V
3. MCB
4. Watthourmeter
5. Kontroler motor kanan
6. Kontroler motor kiri
7. Motor kanan
8. Motor kiri
9. Sensor Sudut

Baterai yang digunakan pada mobil listrik ini baterai Aki dengan tipe VRLA (*Valve-Regulated Lead Acid Battery*) atau yang sering disebut baterai *Maintenance Free* yang memiliki keunggulan bebas dalam perawatan. Mobil listrik yang digunakan dalam penelitian ini memiliki penggerak dua motor BLDC dengan merek yang sama dan daya yang sama yaitu sebesar 800 W dengan tipe atau merek Kelly Motor. Untuk menggerakkan motor BLDC ini digunakan *Controller Motor* dengan spesifikasi dapat menggerakkan daya maksimal yaitu 2000W.

Baterai yang dipakai adalah aki kering (*Free Maintenance*) dengan spesifikasi tegangan 12 Volt 35 Ah dan disambungkan seri dengan aki lain, sehingga total dapat diperoleh sumber 48 Volt.

3.5.2 Diagram Kerja

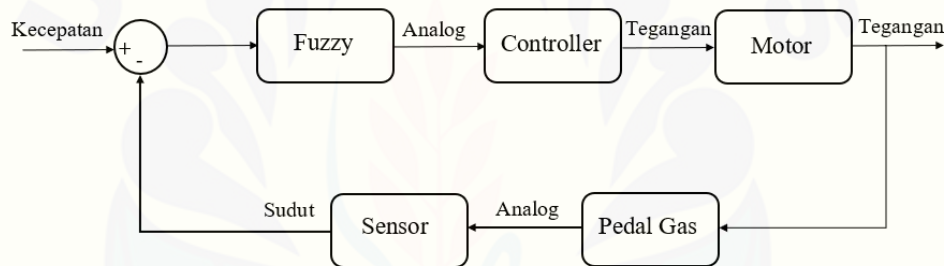


Gambar 3.4 Diagram Kerja

Gambar 3.4 adalah desain elektronik tentang blok diagram mobil listrik dengan penggerak dua motor. Sensor MPU6050 digunakan sebagai pembaca sudut kemudi pada mobil yang akan mengirimkan data tersebut ke mikrokontroler yang akan menentukan kecepatan motor kanan dan motor kiri. Pedal/*throttle* gas digunakan sebagai menentukan batas kecepatan maksimal motor dengan mengirimkan data pada mikrokontroler. Dari mikrokontroler akan mengirimkan data berupa PWM pada motor *controller*, dari motor *controller* mengirimkan berupa tegangan sehingga motor bergerak sesuai perintah. Dalam proses pembuatan pengendali kecepatan mobil listrik berpengerak dua motor dengan metode *Fuzzy* memiliki 3 bagian yaitu *Input*, *Process* dan *Output*. Berikut penjelasan masing – masing bagian :

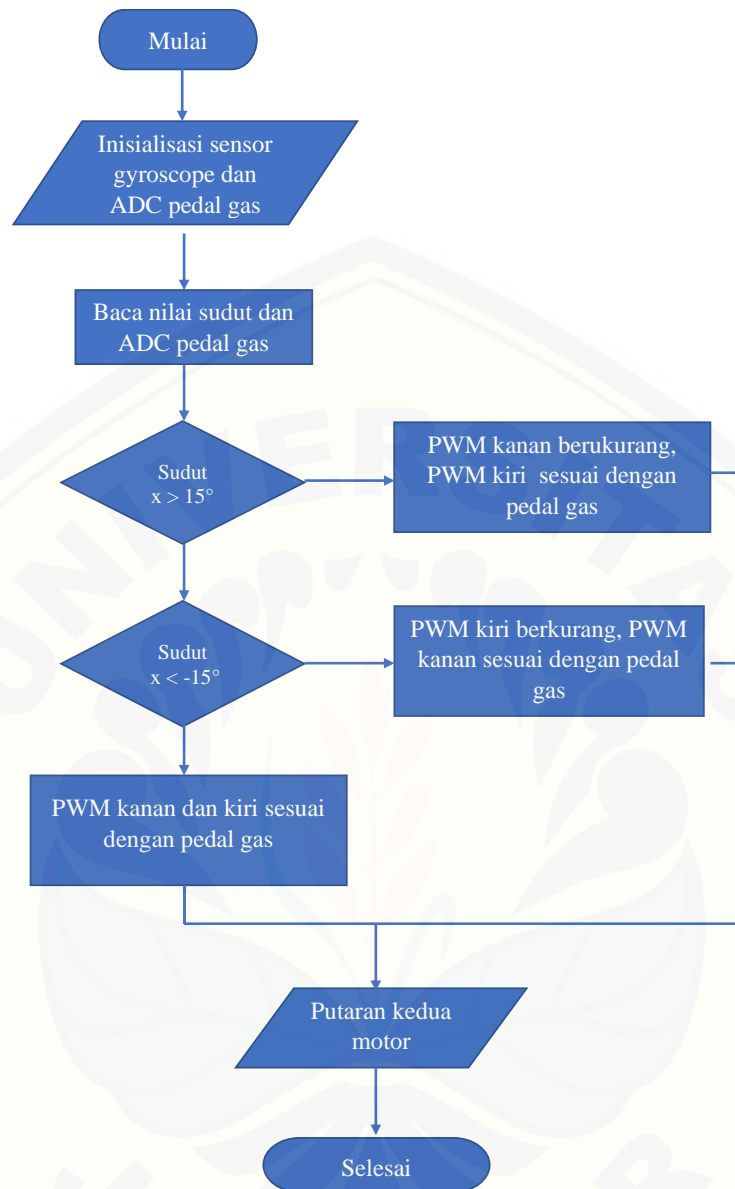
1. *Input*, Bagian *Input* terdiri dari sensor sudut yang berguna sebagai informasi bahwa mobil dalam keadaan berbelok. Kemudian ada pedal / throttle gas sebagai *Input* batas kecepatan maksimal motor.
2. *Process*, Bagian *Process* terdapat mikrokontroler Arduino UNO R3, mikrokontrollere disini sebagai pengendali untuk memproses informasi *Input* menuju *output*. *Process* disini menggunakan metode *Fuzzy* untuk menentukan kecepatan motor kanan dan motor kiri.
3. *Output*, Bagian *Output* terdapat dua motor BLDC. Motor BLDC digunakan sebagai penggerak mobil listrik.

3.5.2 Perancangan Sistem Kontrol



Gambar 3.5 Diagram Blok Sistem Kontrol

Sistem kontrol diatas merupakan respon yang akan diprogramkan oleh mobil, respon pada motor dipengaruhi oleh sensor sudut dan pedal gas dimana motor kanan atau kiri akan berputar dengan kecepatan yang diinginkan sesuai *rule fuzzy* yang telah ditentukan. Pedal gas akan mengirimkan sinyal *Input* berupa analog dan sensor berupa sudut yang akan proses pada mikrokontroler dengan metode *fuzzy*. Mikrokontroler tersebut akan mengirimkan sinyal PWM ke *controller* sesuai dengan *rule fuzzy* dan *controller* mengirimkan tegangan ke motor. Sistem tersebut akan berulang secara *real time* sesuai dengan keadaan mobil.



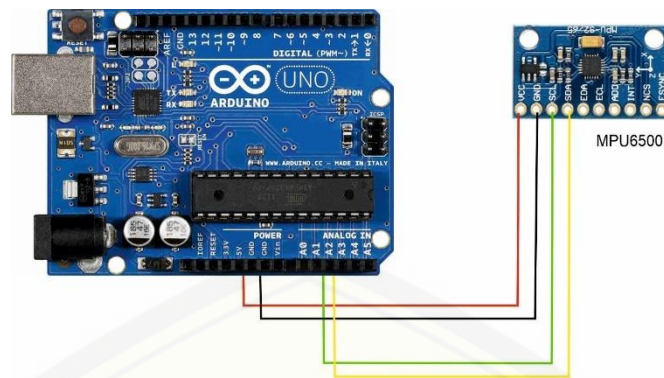
Gambar 3.6 Flowchart Sistem

3.5.3 Perancangan Perangkat Keras Sistem

Pada bagian ini akan dijelaskan berupa perencanaan perangkat keras yang akan digunakan.

a. Rangkaian Sensor Sudut dengan Mikrokontroller

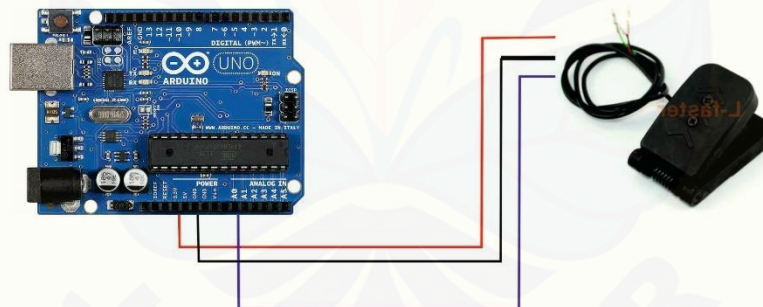
Sensor MPU6050 membaca kemiringan sudut berdasarkan data dari sensor *gyroscope* dan sensor *accelerometer*. Jalur data yang digunakan pada sensor ini adalah jalur data I2C.



Gambar 3.7 Sensor MPU6500 dengan Arduino

b. Rangkaian Pedal Gas dengan Mikrokontroler

Pedal / *throttle* gas ini sebagai *Input* berupa data analog kedalam mikrokontroler, semakin *throttle* gas diputar ke arah maksimum maka putaran atau kecepatan motor semakin bertambah. *Throttle* gas memiliki 3 pin yaitu 5V, Ground, dan *signal*



Gambar 3.8 Pedal Gas Dengan Arduino

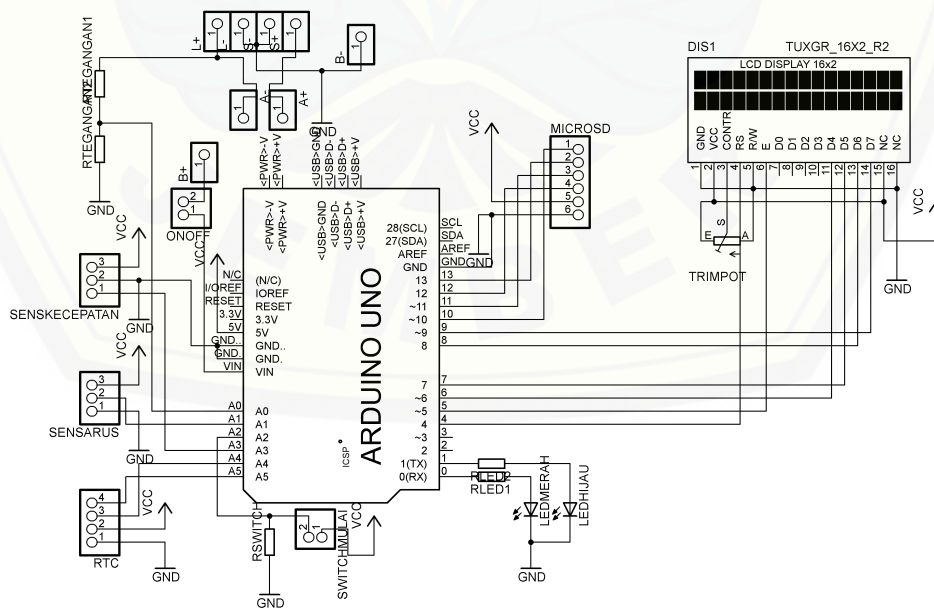
c. Rangkaian Sistem Kontrol

Rangkaian ini menggunakan dua *Input* yaitu pedal gas dan sensor *gyroscope*, pedal mengirimkan sinyal berupa sinyal analog dan sensor *gyroscope* berupa nilai sudut kemudi. Kedua masukan tersebut dikirim ke Arduino UNO R3 yang kemudian di program secara system *fuzzy* menghasilkan keluaran pada mikrokontroler berupa sinyal PWM dengan range tegangan 0 V sampai 5 V yang kemudian dikirim ke *controller* motor.

3.5.4 Perancangan Data Logger

Data Logger digunakan sebagai pembacaan nilai tegangan, arus, dan nilai rpm pada setiap motor kanan dan motor kiri secara real-time ketika mobil listrik berjalan. Pada data logger ini menggunakan sensor tegangan berupa dua resistor yang dirangkai secara seri dengan konsep pembagi tegangan. Nilai resistor yang digunakan sebesar 1k ohm dan 10k Ohm dengan *Input* tegangan maksimal 52V maka nilai keluaran dari pembagi tegangan tersebut sebesar 4,7 V, dimana tegangan tersebut masih mencapai batas aman untuk Arduino Uno. Dalam pembacaan nilai arus menggunakan sensor ACS 712 dengan tipe arus maksimal sebesar 30 A. Digunakannya sensor arus dengan maksimal 30 A ini untuk jaga-jaga jika arus yang dihasilkan ketika pengujian dinamis atau mobil berjalan mencapai 20 A lebih.

Dalam penelitian ini menggunakan dua data logger yang diletakkan pada setiap motor kanan dan motor kiri. Penempatan data logger ini tepat setelah baterai yang kemudian masuk ke *controller* motor. Dalam menyimpan data tersebut digunakan *module* SD Card adaptor dan penambahan RTC untuk penyimpanan dengan waktu secara real-time. Data yang dihasilkan berupa data dengan format “.txt” yang disimpan pada micro SD card. Berikut desain dari rangkaian data logger:



Gambar 3.9 Rangkaian *Data Logger*

3.6 Sistem Kendali *Fuzzy*

Pada Penelitian ini metode *fuzzy* yang digunakan adalah metode Mamdani, keluaran dari metode ini adalah berupa himpunan *fuzzy*. Penelitian ini akan dilakukan pengendalian nilai PWM masukan terhadap kontrol motor yang kemudian akan mempengaruhi kecepatan motor. Set point pada kontrol ini adalah memberi nilai PWM yang diinginkan dan penggunaan sensor *Gyroscope* adalah parameter feedback yang akan mempengaruhi nilai PWM pada keluaran kontrol.

Dalam perancangan suatu pengendali *fuzzy* hal pertama yang dilakukan yaitu menentukan beberapa parameter masukan yang diinginkan, kemudian menentukan parameter keluaran sesuai yang diinginkan. Masukan dari pengendali *Fuzzy* ini adalah nilai ADC pedal gas dan nilai sudut pada sensor, dimana keluaran dari *fuzzy* ini adalah penambahan dan pengurangan PWM yang masuk pada kontrol motor. Berikut fungsi keanggotaan *fuzzy* yang digunakan pada penelitian ini.

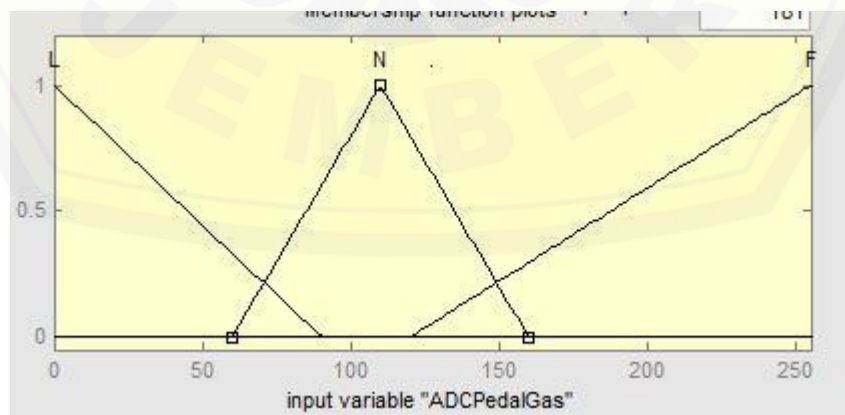
3.6.1 Fungsi Keanggotaan *Fuzzy*

Keanggotaan pada masukan variabel pedal gas dibagi menjadi 3 bagian yaitu sebagai berikut.

L : Low (Rendah)

N : Normal (Sedang)

F : Full (Penuh)



Gambar 3.10 Fungsi Keanggotaan Variabel Pedal Gas

Sedangkan untuk keanggotaan pada masukan variabel sudut kemudi dibagi menjadi 5 bagian, berikut variabel yang digunakan dalam membership function sudut kemudi.

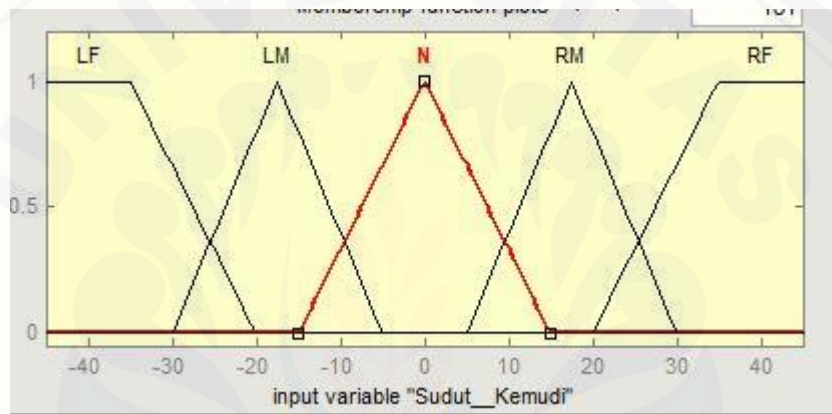
LF : Left Full (Kiri Penuh)

LM : Left Medium (Kiri Sedang)

N : Normal

RN : Right Medium (Kanan Sedang)

RF : Right Full (Kanan Penuh)



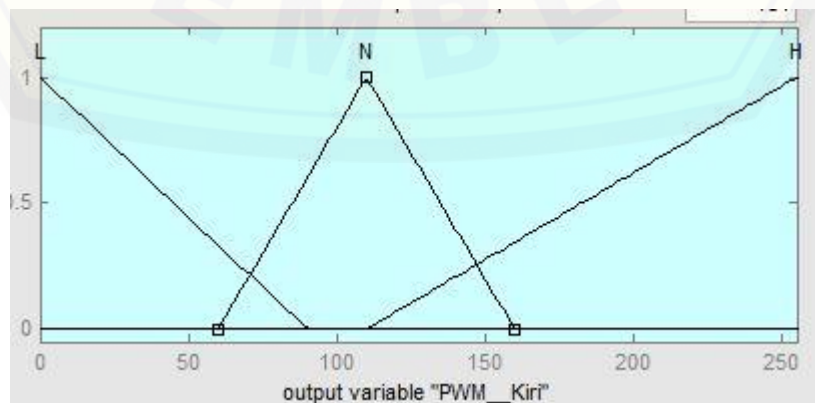
Gambar 3.11 Fungsi Keanggotaan Variabel Sudut Kemudi

Membership function dari keluran nilai PWM setiap motor kiri dan kanan memiliki himpunan yang sama yaitu sebagai berikut:

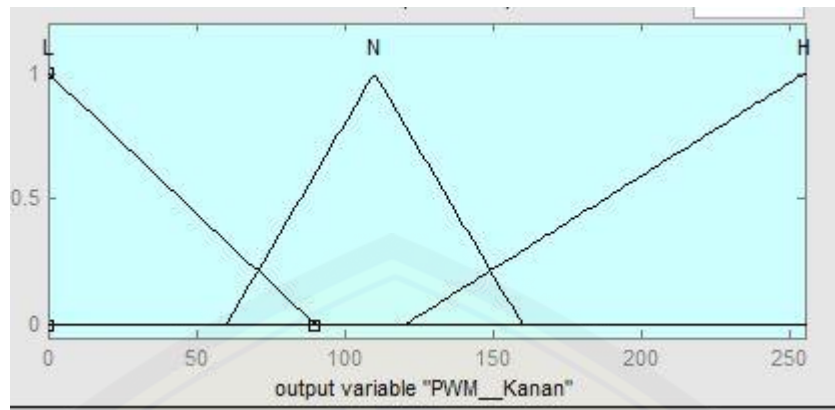
L : Low (Rendah)

N : Normal (Sedang)

F : Full (Penuh)



Gambar 3.12 Fungsi Keanggotaan PWM Motor Kiri



Gambar 3.13 Fungsi Keanggotaan PWM Motor Kanan

3.6.2 Fuzzy Rule Mobil Listrik

Pada umumnya aturan dalam *fuzzy* mempunyai beberapa logika yaitu if dan then dengan nilai masukan yang dikombinasi dengan operator AND ataupun OR. Dari keseluruhan logika tersebut digunakan untuk menentukan keluaran pada *fuzzy rule*.

Tabel 3.2 Fuzzy Rule

Input	Variabel Sudut Kemudi					
	Nilai	LF	LM	N	RM	RF
Variabel Pedal gas	L	LL, RL	LL, RL	LL, RL	LL, RL	LL, RL
	N	LL, RN	LL, RN	LN, RN	LN, RL	LN, RL
	F	LN, RF	LN, RF	LF, RF	LF, RN	LF, RN

- Keterangan :
- LL = PWM Motor Kiri Rendah
 - RL = PWM Motor Kanan Rendah
 - LN = PWM Motor Kiri Sedang
 - RN = PWM Motor Kanan Sedang
 - LF = PWM Motor Kiri Penuh
 - RF = PWM Motor Kanan Penuh

Dalam pengambilan keputusan keanggotaan tiap-tiap fungsi keanggotaan himpunan *fuzzy* masukan ke dalam basis aturan yang telah ditetapkan. Serta *fuzzy*

rule yang dibuat untuk mengendalikan mobil listrik untuk basis aturan *output* berupa PWM dari masukan nilai sudut kemudi dan nilai pedal gas.

3.7 Metode Pengujian dan Pengambilan Data

Metode pengujian dan pengambilan data ini meliputi beberapa pengujian yaitu kalibrasi sensor pada data logger, kalibrasi dan pengujian sensor *Gyroscope* MPU6050, pengujian motor BLDC, pengujian system *fuzzy*, dan pengujian keseluruhan untuk mengetahui konsumsi energi yang digunakan mobil listrik ketika menggunakan sistem kontrol kecepatan dan tanpa menggunakan sistem kontrol kecepatan.

3.7.1 Kalibrasi dan Pengujian Sensor

Pengujian sensor dilakukan guna untuk mengkalibrasi sensor sehingga pembacaan sensor dapat akurat sesuai dengan alat ukur. Pengambilan data kalibrasi sensor ini juga untuk mengetahui kelayakan dari sensor tersebut.

3.7.2 Pengujian Sensor *Gyroscope*

Pada pengujian sensor *Gyroscope* yaitu menguji nilai sudut atau kemiringan yang terbaca pada sensor modul. Nilai yang terbaca dari sensor *Gyroscope* tersebut kemudian dibandingkan dengan nilai kemiringan sudut yang didapatkan dari pengukuran menggunakan busur atau penggaris sudut

3.7.3 Pengujian Sinyal PWM (*Pulse Width Modulation*)

Pengujian dilakukan mikrokontroler atau arduino yaitu proses merubah nilai ADC yang berasal dari *set-point* menjadi keluaran PWM (*pulse width modulation*) pada Arduino Uno R3. Pengujian ini dilakukan untuk mengetahui bagaimana karakteristik sinyal, periode, dan *duty cycle* yang dikeluarkan pada pin *Pulse Width Modulation* pada mikrokontroler yang digunakan dengan frekuensi tertentu.

3.7.4 Pengujian RPM Motor BLDC

Pengujian motor BLDC dilakukan dengan membandingkan kecepatan atau rpm dari kedua motor kanan dan motor kiri. Pengujian ini dilakukan bertujuan mengetahui kecepatan kedua motor ketika dimasukkan nilai PWM yang sama. Pengukuran kecepatan kedua motor diukur menggunakan *tachometer*.

3.7.5 Pengujian Statis dengan Sistem Kontrol Pada Mobil listrik

Pengujian sistem *Fuzzy* dilakukan untuk mengetahui bagaimana pembacaan pedal gas dan sensor *Gyroscope* terhadap keluaran nilai PWM dan kemudian menghasilkan kecepatan pada motor pada mobil listrik. Pengujian ini dilakukan secara statis atau mobil dalam keadaan diam.

3.7.6 Pengujian Keseluruhan

Pengujian keseluruhan ini yaitu mobil listrik yang telah diinstalasi secara keseluruhan dengan sistem kontrol yang telah dibuat. Pengujian ini dilakukan secara dinamis atau mobil listrik berjalan melalui jalur yang telah ditentukan. Dari hasil pengujian keseluruhan sistem tersebut akan menghasilkan perbandingan penggunaan energi pada mobil listrik ketika menggunakan sistem kontrol dan tanpa menggunakan sistem kontrol.

BAB 5. PENUTUP

5.1 Kesimpulan

Berdasarkan data pengujian mobil listrik berpengerak dua motor dengan pengaturan kecepatan menggunakan sensor *gyroscope* yang telah dilakukan, diperoleh kesimpulan yaitu:

1. Dari pengujian kecepatan atau rpm motor secara statis menggunakan sistem dengan PWM pedal gas 140, ketika kemudi lurus kecepatan kedua motor hampir sama yaitu 317,9 rpm dan 319 rpm, sedangkan ketika kemudi berbelok kecepatan salah satu motor menurun yaitu dengan nilai rata-rata 117,11 rpm.
2. Pada pengujian dinamis ketika mobil tanpa menggunakan sistem kontrol dengan sensor *gyroscope*, nilai arus motor yang dijadikan poros mencapai 5,9A ketika berbelok yang mengakibatkan penggunaan daya semakin meningkat.
3. Pada pengujian dinamis yang telah dilakukan ketika mobil listrik menggunakan sistem kontrol dengan sensor *gyroscope*, nilai arus motor yang dijadikan poros saat berbelok hanya mencapai 1,4 A yang mengakibatkan penggunaan daya menurun.
4. Pada pengujian penggunaan energi dengan lintasan searah jarum jam, jarak tempuh per satuan kwh mobil listrik ketika menggunakan sistem kontrol dengan sensor *gyroscope* mencapai nilai sebesar 67,11 km/kwh, lebih efisien dibandingkan mobil listrik tanpa menggunakan sistem kontrol *gyroscope* yaitu sebesar 60,06 km/kwh.
5. Pada pengujian penggunaan energi dengan lintasan berlawanan arah jarum jam, jarak tempuh per satuan kwh mobil listrik ketika menggunakan sistem kontrol dengan sensor *gyroscope* menghasilkan nilai sebesar 71,68 km/kwh, hasil ini juga lebih efisien dibandingkan mobil listrik tanpa menggunakan sistem kontrol *gyroscope* yaitu sebesar 63,09 km/kwh

5.2 Saran

Berdasarkan data pengujian mobil listrik berpengerak dua motor dengan pengaturan kecepatan menggunakan sensor *gyroscope* yang telah dilakukan, disarankan bahwa:

1. Penggantian mikrokontroler dengan spesifikasi yang ditingkatkan atau respon yang lebih maksimal seperti ARM.
2. Desain *fuzzy logic* lebih halus atau lebih baik dengan *rule base* yang bervariasi.
3. Pertimbangan dalam segi mekanik mobil karena sangat berpengaruh dalam pengujian secara dinamis.
4. Menggunakan jenis baterai yang lebih bagus seperti baterai Li-po ataupun Li-ion yang memiliki nilai discharge yang lebih tinggi.
5. Dapat dikembangkan untuk peneliatian selanjutnya yaitu kendaraan otonom atau kendaraan otomatis dengan penampahan acktuator pada kemudi yang dilengkapi pembacaan GPS sebagai pengatur arah jalannya kendaraan.
6. Disarankan untuk penelitian selanjutnya agar kedua motor sebagai pengerak diletakkan di depan.

DAFTAR PUSTAKA

- Amanda, I., I. Hidayat, A. Rusdinar, F. T. Elektro, U. Telkom, dan J. Telekomunikasi. 2015. Kendali kecepatan mobil listrik menggunakan dua motor listrik dengan *fuzzy-pid control speed of electric car uses two electric motors with fuzzy-pid*. 2(3):6960–6968.
- Chritian, F. L. dan M. R. Nugraha. 2013. Pengendali kecepatan motor dc *Brushless* menggunakan *microcontroller avr*.
- InvenSense Inc. 2014. MPU-6500 product specification revision 1.1. 1(408):1–40.
- Ismail, A. K. 2019. Rancang bangun sistem pengendali kecepatan motor *Brushless Direct Current* pada kendaraan listrik menggunakan metode *fuzzy inference system*.
- Kandiban, R. dan R. Arulmozhiyal. 2012. *Speed control of bldc motor using adaptive fuzzy pid controller*. *Procedia Engineering*. 38:306–313.
- Kelly Controllers. 2012. Kelly kbl *Brushless motor controller user 's manual*. 12.
- Kristiyono, R., O. Wahyunggoro, dan P. Nugroho. 2015. Sistem kendali kecepatan motor bldc menggunakan algoritma hybrid pid *fuzzy*. 116–124.
- Liun, E. 2018. Dampak peralihan massal transportasi jalan raya ke mobil listrik. 19(2):113–122.
- Malik, I. A. dan N. Hariyanto. 2013. Analisis penghematan energi motor listrik di pt . x. 1(3):281–294.
- Masudi, N. 2014. Desain *controller* motor bldc untuk meningkatkan performa (daya *output*) sepeda motor listrik design of bldc motor *controller* for increasing the *output* performance (*output power*) from electric bike. 1–65.
- Mpc, C. 2006. BLDC motor with *speed* closed loop and dc-bus break *controller* , driven by etpu on mpc5554. 1–50.
- Nugroho, S. A., M. A. Murti, dan A. Rusdinar. 2010. Perancangan dan implementasi dc to dc converter sebagai pulse width modulation.
- Pindoriya, R., S. Rajendran, dan P. Chauhan. 2014. *Speed control* of bldc motor using sinusoidal pwm technique. *Ijaerd.Co.In*. (March):1–6.
- Sukerayasa, I. W. 2009. Nyoman s kumara , i wayan sukerayasa. *Tinjau*

Perkembangan Kendaraan ;Istrik Dunia Hingga Sekarang. 8

Tahami, F., S. Farhangi, dan R. Kazemi. 2004. A *fuzzy logic Direct yaw-moment control* system for all-wheel-drive electric vehicles. *Vehicle System Dynamics*. 41(3):203–221.

Wicaksono, A. S., R. E. A. K, dan E. Iskandar. 2016. Perancangan dan implementasi sistem pengaturan kecepatan motor bldc menggunakan kontroler pi berbasis neural-*fuzzy* hibrida adaptif. 5(2):68–74.



LAMPIRAN**A. Listing Program Pada Mikrokontroler**

```
#include <Wire.h>
#include <SimpleKalmanFilter.h>
#include <LiquidCrystal.h>
#include "fis_header.h"

LiquidCrystal lcd(4, 5, 6, 7, 8, 9);

// 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 = 2;
// Number of rules to the fuzzy inference system
const int fis_gcR = 15;

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

int POT = A0;
int PWM1=10;
int PWM2= 11;
int nilai;
int output1, output2, motor1, motor2;
int addpwm = 0;

//Gyro Variables
float elapsedTime, time, timePrev;
int gyro_error=0;
float Gyr_rawX, Gyr_rawY, Gyr_rawZ;
float Gyro_angle_x, Gyro_angle_y;
float Gyro_raw_error_x, Gyro_raw_error_y;

//Acc Variables
int acc_error=0;
float rad_to_deg = 180/3.141592654;
float Acc_rawX, Acc_rawY, Acc_rawZ;
float Acc_angle_x, Acc_angle_y;
float Acc_angle_error_x, Acc_angle_error_y;
float Total_angle_x, Total_angle_y;

//nilai tuning kalman filter
SimpleKalmanFilter simpleKalmanFilterx(2, 2, 0.01);
SimpleKalmanFilter simpleKalmanFiltery(2, 2, 0.01);
```

```
// Serial output refresh time
const long SERIAL_REFRESH_TIME = 100;
long refresh_time;
float nilai_sudutx;
float nilai_suduty;
float kalibrasix;

void setup() {

  lcd.begin(16, 2);
  pinMode (POT, INPUT);
  pinMode (PWM1, OUTPUT); // Pin mode for Output: PWM_Kiri
  pinMode (PWM2, OUTPUT); // Pin mode for Output:
  PWM_Kanan

  Wire.begin();
  Wire.beginTransmission(0x68);
  Wire.write(0x6B);
  Wire.write(0x00);
  Wire.endTransmission(true);
  //Gyro config
  Wire.beginTransmission(0x68);
  Wire.write(0x1B);
  Wire.write(0x10);
  Wire.endTransmission(true);
  //Acc config
  Wire.beginTransmission(0x68);
  Wire.write(0x1C);
  Wire.write(0x10);
  Wire.endTransmission(true);

  Serial.begin(9600);
  time = millis();

  /*Here we calculate the acc data error before we start
  the loop
  * I make the mean of 200 values, that should be enough*/
  if(acc_error==0)
  {
    for(int a=0; a<200; a++)
    {
      Wire.beginTransmission(0x68);
      Wire.write(0x3B);
      Wire.endTransmission(false);
      Wire.requestFrom(0x68, 6, true);
```

```

    Acc_rawX=(Wire.read()<<8|Wire.read())/4096.0    ;
//each value needs two registres
    Acc_rawY=(Wire.read()<<8|Wire.read())/4096.0 ;
    Acc_rawZ=(Wire.read()<<8|Wire.read())/4096.0 ;

    /*---X---*/
    Acc_angle_error_x = Acc_angle_error_x +
    ((atan((Acc_rawY)/sqrt(pow((Acc_rawX),2) +
    pow((Acc_rawZ),2)))*rad_to_deg));
    /*---Y---*/
    Acc_angle_error_y = Acc_angle_error_y + ((atan(-
    1*(Acc_rawX)/sqrt(pow((Acc_rawY),2) +
    pow((Acc_rawZ),2)))*rad_to_deg));

    if(a==199)
    {
        Acc_angle_error_x = Acc_angle_error_x/200;
        Acc_angle_error_y = Acc_angle_error_y/200;
        acc_error=1;
    }
}
} //end of acc error calculation

/*Here we calculate the gyro data error before we start
the loop
* I make the mean of 200 values, that should be enough*/
if(gyro_error==0)
{
    for(int i=0; i<200; i++)
    {
Wire.beginTransmission(0x68);
Wire.write(0x43);
Wire.endTransmission(false);
Wire.requestFrom(0x68,4,true);

Gyr_rawX=Wire.read()<<8|Wire.read();
Gyr_rawY=Wire.read()<<8|Wire.read();
    /*---X---*/
Gyro_raw_error_x = Gyro_raw_error_x + (Gyr_rawX/32.8);
    /*---Y---*/
    Gyro_raw_error_y = Gyro_raw_error_y +
(Gyr_rawY/32.8);
    if(i==199)
    {
        Gyro_raw_error_x = Gyro_raw_error_x/200;
        Gyro_raw_error_y = Gyro_raw_error_y/200;
        gyro_error=1;
    }
}
}

```

```

    }
  }
  } //end of gyro error calculation
} //end of setup void

Void FIS

void fis
{
  // Read Input: ADCPedalGas
  g_fisInput[0] = analogRead(0);
  // Read Input: Sudut_Kemudi
  g_fisInput[1] = analogRead(1);

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

  fis_evaluate();

  // Set output vlaue: PWM_Kiri
  analogWrite(2 , g_fisOutput[0]);
  // Set output vlaue: PWM_Kanan
  analogWrite(3 , g_fisOutput[1]);

  ////////////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[] = { 3, 5 };

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

    // Coefficients for the Input Member Functions
    FIS_TYPE  fis_gMFI0Coeff1[] = { 60.6746031746032,
    110.674603174603, 160.674603174603 };
    FIS_TYPE  fis_gMFI0Coeff2[] = { 120, 255, 255 };
    FIS_TYPE  fis_gMFI0Coeff3[] = { 0, 0, 90 };
    FIS_TYPE*  fis_gMFI0Coeff[] = { fis_gMFI0Coeff1,
    fis_gMFI0Coeff2, fis_gMFI0Coeff3 };
    FIS_TYPE  fis_gMFI1Coeff1[] = { -45, -45, -35, -20 };
    FIS_TYPE  fis_gMFI1Coeff2[] = { -15, 0, 15 };

```

```

FIS_TYPE fis_gMFI1Coeff3[] = { 20, 35, 45, 45 };
FIS_TYPE fis_gMFI1Coeff4[] = { 5, 17.5, 30 };
FIS_TYPE fis_gMFI1Coeff5[] = { -30, -17.5, -5 };
FIS_TYPE* fis_gMFI1Coeff[] = { fis_gMFI1Coeff1,
fis_gMFI1Coeff2, fis_gMFI1Coeff3, fis_gMFI1Coeff4,
fis_gMFI1Coeff5 };
FIS_TYPE** fis_gMFI1Coeff[] = { fis_gMFI1Coeff,
fis_gMFI1Coeff };

// Coefficients for the Output Member Functions
FIS_TYPE fis_gMFO0Coeff1[] = { 0, 0, 90 };
FIS_TYPE fis_gMFO0Coeff2[] = { 60, 110, 160 };
FIS_TYPE fis_gMFO0Coeff3[] = { 110, 255, 255 };
FIS_TYPE* fis_gMFO0Coeff[] = { fis_gMFO0Coeff1,
fis_gMFO0Coeff2, fis_gMFO0Coeff3 };
FIS_TYPE fis_gMFO1Coeff1[] = { 0, 0, 90 };
FIS_TYPE fis_gMFO1Coeff2[] = { 60, 110, 160 };
FIS_TYPE fis_gMFO1Coeff3[] = { 120, 255, 255 };
FIS_TYPE* fis_gMFO1Coeff[] = { fis_gMFO1Coeff1,
fis_gMFO1Coeff2, fis_gMFO1Coeff3 };
FIS_TYPE** fis_gMFO1Coeff[] = { fis_gMFO1Coeff,
fis_gMFO1Coeff };

// Input membership function set
int fis_gMFI0[] = { 0, 0, 0 };
int fis_gMFI1[] = { 1, 0, 1, 0, 0 };
int* fis_gMFI[] = { fis_gMFI0, fis_gMFI1};

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

//////////////////////          RULE          BASE          FUZZY
//////////////////////
// Rule Weights
FIS_TYPE fis_gRWeight[] = { 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1 };
// Rule Type
int fis_gRType[] = { 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1 };

// Rule Inputs
int fis_gRI0[] = { 2, 2 };
int fis_gRI1[] = { 3, 2 };
int fis_gRI2[] = { 1, 2 };
int fis_gRI3[] = { 3, 1 };

```

```
int fis_gRI4[] = { 3, 3 };
int fis_gRI5[] = { 3, 4 };
int fis_gRI6[] = { 3, 5 };
int fis_gRI7[] = { 1, 1 };
int fis_gRI8[] = { 1, 5 };
int fis_gRI9[] = { 1, 3 };
int fis_gRI10[] = { 1, 4 };
int fis_gRI11[] = { 2, 4 };
int fis_gRI12[] = { 2, 3 };
int fis_gRI13[] = { 2, 5 };
int fis_gRI14[] = { 2, 1 };
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 };

// Rule Outputs
int fis_gRO0[] = { 3, 3 };
int fis_gRO1[] = { 1, 1 };
int fis_gRO2[] = { 2, 2 };
int fis_gRO3[] = { 1, 1 };
int fis_gRO4[] = { 1, 1 };
int fis_gRO5[] = { 1, 1 };
int fis_gRO6[] = { 1, 1 };
int fis_gRO7[] = { 1, 2 };
int fis_gRO8[] = { 1, 2 };
int fis_gRO9[] = { 2, 1 };
int fis_gRO10[] = { 2, 1 };
int fis_gRO11[] = { 3, 2 };
int fis_gRO12[] = { 3, 2 };
int fis_gRO13[] = { 2, 3 };
int fis_gRO14[] = { 2, 3 };
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 };

// Input range Min
FIS_TYPE fis_gIMin[] = { 0, -45 };

// Input range Max
FIS_TYPE fis_gIMax[] = { 255, 45 };

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

// Output range Max
```

```

FIS_TYPE fis_gOMax[] = { 255, 255 };

//*****
// 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 = (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);
}

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

```

```

    int i;

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

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

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

    // Transforming input to fuzzy Input
    int i, j, r, o;
    for (i = 0; i < fis_gcI; ++i)
    {
        for (j = 0; j < fis_gIMFCount[i]; ++j)
        {
fuzzyInput[i][j] =
            (fis_gMF[fis_gMFI[i]
[j]])(g_fisInput[i], fis_gMFICoeff[i][j]);
        }
    }

    int index = 0;

```

```

for (r = 0; r < fis_gcR; ++r)
{
    if (fis_gRType[r] == 1)
    {
        fuzzyFires[r] = FIS_MAX;
        for (i = 0; i < fis_gcI; ++i)
        {
            index = fis_gRI[r][i];
            if (index > 0)
                fuzzyFires[r] =
                fis_min(fuzzyFires[r], fuzzyInput[i][index - 1]);
            else if (index < 0)
                fuzzyFires[r] = fis_min(fuzzyFires[r], 1 - fuzzyInput[i]
                [-index - 1]);
            else
                fuzzyFires[r] = fis_min(fuzzyFires[r], 1);
        }
    }
    else
    {
        fuzzyFires[r] = FIS_MIN;
        for (i = 0; i < fis_gcI; ++i)
        {
            index = fis_gRI[r][i];
            if (index > 0)
                fuzzyFires[r] =
                fis_max(fuzzyFires[r],
                fuzzyInput[i][index - 1]);
            else if (index < 0)
                fuzzyFires[r] =
                fis_max(fuzzyFires[r], 1 -
                fuzzyInput[i][-index - 1]);
            else
                fuzzyFires[r] = fis_max(fuzzyFires[r], 0);
        }
    }
    fuzzyFires[r] = fis_gRWeight[r] * fuzzyFires[r];
    sW += fuzzyFires[r];
}
if (sW == 0)
{
    for (o = 0; o < fis_gcO; ++o)
    {
        g_fisOutput[o] = ((fis_gOMax[o] + fis_gOMin[o]) /
        2);
    }
}
else
{

```



```
        for (o = 0; o < fis_gc0; ++o)
        {
            g_fisOutput[o]
fis_defuzz_centroid(fuzzyRuleSet, o);
        }
    }
}

void loop()
{
    kalman();
    fis ();
    nilai = analogRead(POT);
    output1 = map(nilai, 0, 1023, 0, 255);
    output2 = map(nilai, 0, 1023, 0, 255);
    addpwm = constrain(addpwm,0,255);

    Serial.print(kalibrasix);
    Serial.print(" | ");
    Serial.print(motor1);
    Serial.print(" | ");
    Serial.print(motor2);
    Serial.print(" | ");
    Serial.println(addpwm);
    lcd.setCursor(0, 0);
    lcd.print(motor1);
    lcd.setCursor(6, 0);
    lcd.print(" || ");

    lcd.setCursor(13, 0);
    lcd.print(motor2);

    lcd.setCursor(2, 1);
    lcd.print("sudut:");
    lcd.setCursor(9, 1);
    lcd.print(kalibrasix);
    lcd.setCursor(13, 1);
    lcd.print("dr");

    if (kalibrasix < -20)
    {
        motor1 = 0 ;
        motor2 = output2;
        addpwm = 0;
        analogWrite (PWM1, motor1);
    }
}
```

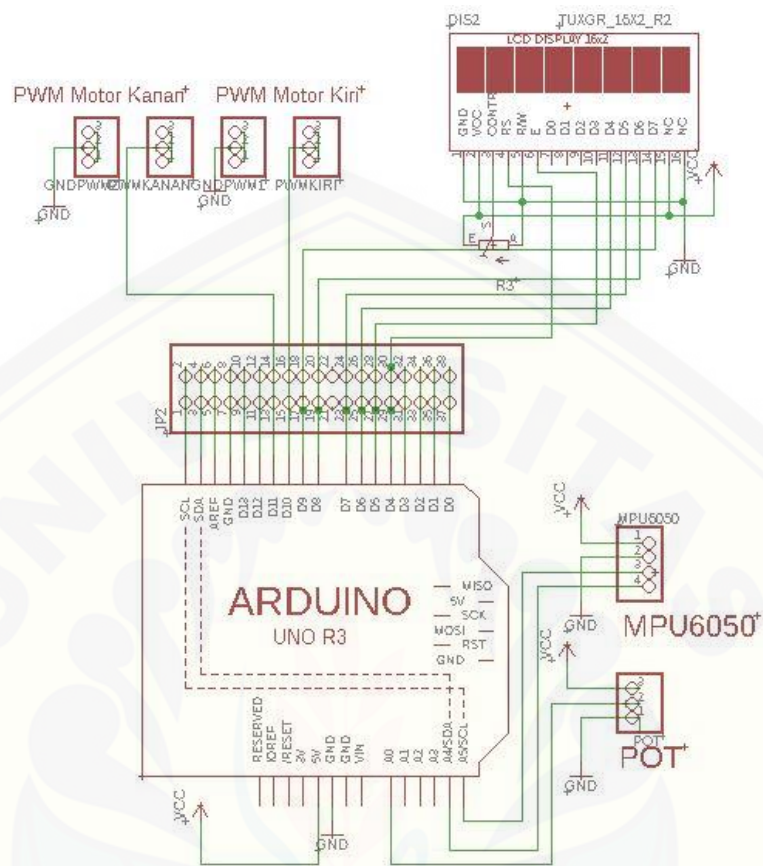
```
        analogWrite (PWM2, motor2);

    }
    else if (kalibrasix > 20)
    {
        motor1 = output1 ;
        motor2 = 0 ;
        addpwm = 0;
        analogWrite (PWM1, motor1);
        analogWrite (PWM2, motor2);
    }
    // else
    // {
    //     motor1 = output1;
    //     motor2 = output2;
    //     analogWrite (PWM1, output1);
    //     analogWrite (PWM2, output2);
    // }

    else
    {
        motor1 = addpwm;
        motor2 = addpwm;
        addpwm = addpwm+2;
        analogWrite(PWM1,motor1);
        analogWrite (PWM2,motor2);

        if (addpwm >= 120)
        {
            motor1 = output1;
            motor2 = output2;
            analogWrite(PWM1,motor1);
            analogWrite (PWM2, motor2);

        }
        delay(100);
    }
}
```



Gambar Rangkaian Sistem Pada Software Eagle