

**PENGATURAN ARAH DAN KECEPATAN PUTAR DISK
INERSIA UNTUK MENJAGA KESEIMBANGAN MINIATUR
SEPEDA MOTOR MENGGUNAKAN KONTROL
LOGIKA FUZZY**

SKRIPSI

JURUSAN TEKNIK ELEKTRO

Diajukan Untuk Memenuhi Persyaratan

Memperoleh Gelar Sarjana Teknik



Disusun oleh :

ADHITYA YOGA GURITNA

NIM. 0810633001

**KEMENTERIAN PENDIDIKAN NASIONAL
UNIVERSITAS BRAWIJAYA
FAKULTAS TEKNIK
MALANG
2012**

LEMBAR PERSETUJUAN
PENGATURAN ARAH DAN KECEPATAN PUTAR DISK INERSIA
UNTUK MENJAGA KESEIMBANGAN MINIATUR SEPEDA MOTOR
MENGGUNAKAN KONTROL LOGIKA FUZZY

SKRIPSI
KONSENTRASI TEKNIK KONTROL

Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik



Disusun oleh:
ADHITYA YOGA G
NIM. 0810633001

Telah diperiksa dan disetujui oleh :

Dosen Pembimbing I

Dosen Pembimbing II

Tri Nurwati. ST., MT

NIP. 19790615 200812 2 003

M. Aziz Muslim. ST., MT., Ph.D

NIP. 19741203 200012 1 001

LEMBAR PENGESAHAN
PENGATURAN ARAH DAN KECEPATAN PUTAR DISK INERSIA
UNTUK MENJAGA KESEIMBANGAN MINIATUR SEPEDA MOTOR
MENGGUNAKAN KONTROL LOGIKA FUZZY

SKRIPSI

JURUSAN TEKNIK ELEKTRO

Diajukan Untuk Memenuhi Persyaratan

Memperoleh Gelar Sarjana Teknik

Disusun oleh:

ADHITYA YOGA GURITNA

NIM. 0810633001

Skripsi ini telah diuji dan dinyatakan lulus pada

Tanggal 26 Desember 2012

DOSEN PENGUJI

Ir. Purwanto., MT

NIP. 19540424 198601 1 001

Ir. Retnowati., MT

NIP. 19511224 198203 2 001

Goegoes Dwi N, ST., MT

NIP. 19711013 200604 1 001

Mengetahui

Ketua Jurusan Teknik Elektro

Dr. Ir. Sholeh Hadi Pramono, MS.

NIP. 19580728 198701 1 001

KATA PENGANTAR



Alhamdulillah, segala puji hanya bagi Allâh Subhanahu Wa Taâla, Rabb alam semesta. Dialah Allâh, Tuhan Yang Maha Satu, Yang Maha Pengasih lagi Maha Penyayang. Dialah Sebaik baik Penolong dan Sebaik baik Pelindung. Shalawat dan salâm kepada Nabi Muhammad Rasulullâh Shallallâhu Alaihi Wa Salâm, pemilik syafaât di Hari Akhir, sang pembawa kabar gembira dan sebaik baik suri tauladan bagi yang mengharap Rahmat dan Hidayah-Nya.

Sungguh hanya melalui Pertolongan dan Perlindungan Allâh SWT semata sehingga penulis dapat menyelesaikan skripsi ini. Dengan seizin Allâh SWT, di kesempatan yang baik ini penulis ingin menghaturkan rasa terima kasih dan penghargaan yang sebesar besarnya atas bantuan sehingga terselesainya skripsi ini kepada:

1. DR. Ir. Sholeh Hadi Pramono, MS selaku Ketua Jurusan Teknik Elektro Universitas Brawijaya dan Bapak Muhammad Aziz Muslim, ST., MT., Ph.D. selaku Sekretaris Jurusan Teknik Elektro.
2. Ibu Tri Nurwati. ST., MT dan Bapak Muhammad Aziz Muslim, ST., MT., Ph.D selaku dosen pembimbing skripsi yang telah memberikan masukan, saran, solusi atas segala permasalahan serta kesabaran beliau dalam proses membimbing hingga terselesaikannya skripsi ini.
3. Ibu Tri Nurwati. ST., MT selaku dosen pembimbing akademik.
4. Keluarga tercinta, Ayah, Ibu dan Kakak yang selalu memberi doa, motivasi, dorongan, semangat dan kasih penulis yang sangat besar.
5. Keluarga besar Angkatan 2008 “CONCORDES”. Untuk teman – teman seperjuangan, Katherin, Andika, Arif, Mifta, Okta, Angga, Risa P, Risa A, Ninin, Pribadi, Wiyoga, dan Wiwik terimakasih telah menemani dan membuat suasana pengerjaan skripsi ini terasa menyenangkan.
6. Team Evaluator PROBINMABA 2010/2011. Baskoro, Izat, Jumadil, Firsada, Umar, Budi, Ikhsan, Katherin, Risa, Wiwik, Manda, Yanti, Rosa, dan Mita yang telah membantu menyelesaikan tanggung jawab yang di berikan kepada penulis.

7. Keluarga BEM FT UB 2011-2012. Titin, Aryo, Firsada, Rizal, Nelza, Dirham, Rifqi dan Avent. Pengalaman, canda tawa dan prestasi yang pernah kita ukir dengan sangat indah takkan pernah bisa dilupakan.
8. Bapak , Ibu dosen beserta segenap staf dan karyawan Jurusan Teknik Elektro baik secara langsung maupun tidak langsung yang telah banyak membantu dalam penyelesaian skripsi ini.
9. Ariska Desy Haryani atas segala pelajaran, dorongan, omelan, kesetiaan dan kasih sayang serta cinta yang terlampaui berarti untuk diungkapkan.
10. Basecamp Sengguruh Fullhouse. Jefri, Reza, Aldo, Liki, Limbat, andika dan mifta. Suatu kebanggaan tersendiri pernah tinggal serumah dengan kalian.
11. Semua pihak yang telah memberikan bantuan serta dukungan baik secara langsung maupun tidak langsung atas penyusunan skripsi ini.

Sekiranya Allâh SWT mencatat amalan ikhlas kami dan semua pihak yang turut membantu sehingga skripsi ini terselesaikan. Akhirnya, penulis menyadari bahwa skripsi ini masih jauh dari sempurna namun semoga skripsi ini dapat memberikan manfaat bagi kita semua. Allâhumma Amîn.

Malang, Desember 2012

Penulis

DAFTAR ISI

LEMBAR PERSETUJUAN.....	i
KATA PENGANTAR	iii
DAFTAR ISI.....	v
DAFTAR GAMBAR	ix
DAFTAR TABEL	xi
ABSTRAK.....	xii
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	3
1.5 Manfaat.....	3
1.6 Sistematika Pembahasan	3
BAB II TINJAUAN PUSTAKA.....	5
2.1 Momen Inersia.....	5
2.2 Kontrol Logika <i>Fuzzy</i>	6
2.2.1 Struktur Dasar Kontrol Logika Fuzzy.....	6
2.2.2 Fuzzifikasi.....	7
2.2.3 Aturan Fuzzy (Fuzzy Rule).....	8
2.2.4 Metode Inferensi	9
2.2.4.1 Metode INFERENSI MAX – MIN	9
2.2.4.2 Metode INFERENSI MAX – DOT.....	10
2.2.5 Defuzzifikasi	11
2.3 Accelerometer.....	12
2.4 Motor DC.....	13
2.4.1 Prinsip Kerja Motor DC	13

2.4.2	Pengaturan Motor DC	14
2.5	PWM (Pulse Width Modulation).....	15
2.6	Driver Motor DC	17
2.7	AVR AT Mega 8	18
2.7.1	Arsitektur Mikrokontroler AT Mega 8	19
2.7.2	Fitur AT Mega 8	20
2.7.3	Konfigurasi AT Mega 8	20
2.8	Program Code Vision AVR.....	21
BAB III METODOLOGI.....		24
3.1	Studi literatur	24
3.2	Perancangan Sistem.....	24
3.3	Realisasi Pembuatan Sistem.....	25
3.4	Pengujian dan Analisa Data.....	25
3.5	Pengambilan Kesimpulan.....	26
BAB IV PERANCANGAN DAN PEMBUATAN ALAT		27
4.1	Perancangan Sistem.....	27
4.1.1	Blok Diagram	27
4.1.2	Prinsip Kerja Alat.....	28
4.2	Perancangan Perangkat Keras (Hardware).....	28
4.2.1	Accelerometer	28
4.2.2	Perancangan Driver Motor	29
4.2.3	Rangkaian Mikrokontroler AT Mega 8.....	31
4.2.4	Perancangan Mekanik Miniatur Sepeda Motor.....	33
4.2.4.1	Spesifikasi Alat	35
4.2.4.2	Pemilihan Motor DC	35
4.3	Perancangan Perangkat Lunak (software)	36

4.3.1	Perancangan Kontroler Logika Fuzzy.....	36
4.3.1.1	Menentukan Variabel Masukan dan Variabel Keluaran.....	36
4.3.1.2	Fuzzifikasi.....	36
4.3.1.3	Membership Function Input.....	37
4.3.1.3.1	Membership Function Kemiringan (error (e)).....	37
4.3.1.3.2	Membership Function Perubahan Kemiringan (derror (de/dt))....	37
4.3.1.3.3	Menentukan batas MF pada Kemiringan dan Perubahan Kemiringan	37
4.3.1.3.4	Menentukan Batas MF Keluaran Pada Motor DC	39
4.3.1.4	Kaidah Atur Kontrol Logika Fuzzy	40
4.3.2	Perancangan Algoritma Perangkat Lunak.....	41
BAB V PENGUJIAN DAN ANALISIS SISTEM.....		43
5.1	Pengujian Motor DC.....	43
5.1.1	Tujuan.....	43
5.1.2	Peralatan yang digunakan.....	43
5.1.3	Langkah pengujian.....	44
5.2	Pengujian Driver Motor.....	46
5.2.1	Tujuan	46
5.2.2	Peralatan yang Digunakan.....	46
5.2.3	Langkah Pengujian.....	47
5.3	Pengujian Accelerometer.....	50
5.3.1	Tujuan.....	50
5.3.2	Peralatan yang Digunakan	50
5.3.3	Langkah pengujian.....	50
5.4	Pengujian Titik Berat.....	52
5.4.1	Tujuan.....	52
5.4.2	Peralatan yang Digunakan	52

5.4.3 Langkah Pengujian	52
5.5 Pengujian Keseluruhan.....	53
5.5.1 Tujuan.....	53
5.5.2 Peralatan yang Digunakan	53
5.5.3 Langkah Pengujian	53
5.5.4 Hasil Pengujian Keseluruhan.....	55
BAB VI PENUTUP	59
6.1 Kesimpulan.....	59
6.2 Saran.....	59
DAFTAR PUSTAKA	61
LAMPIRAN I.....	62
LAMPIRAN II.....	65
LAMPIRAN III.....	81



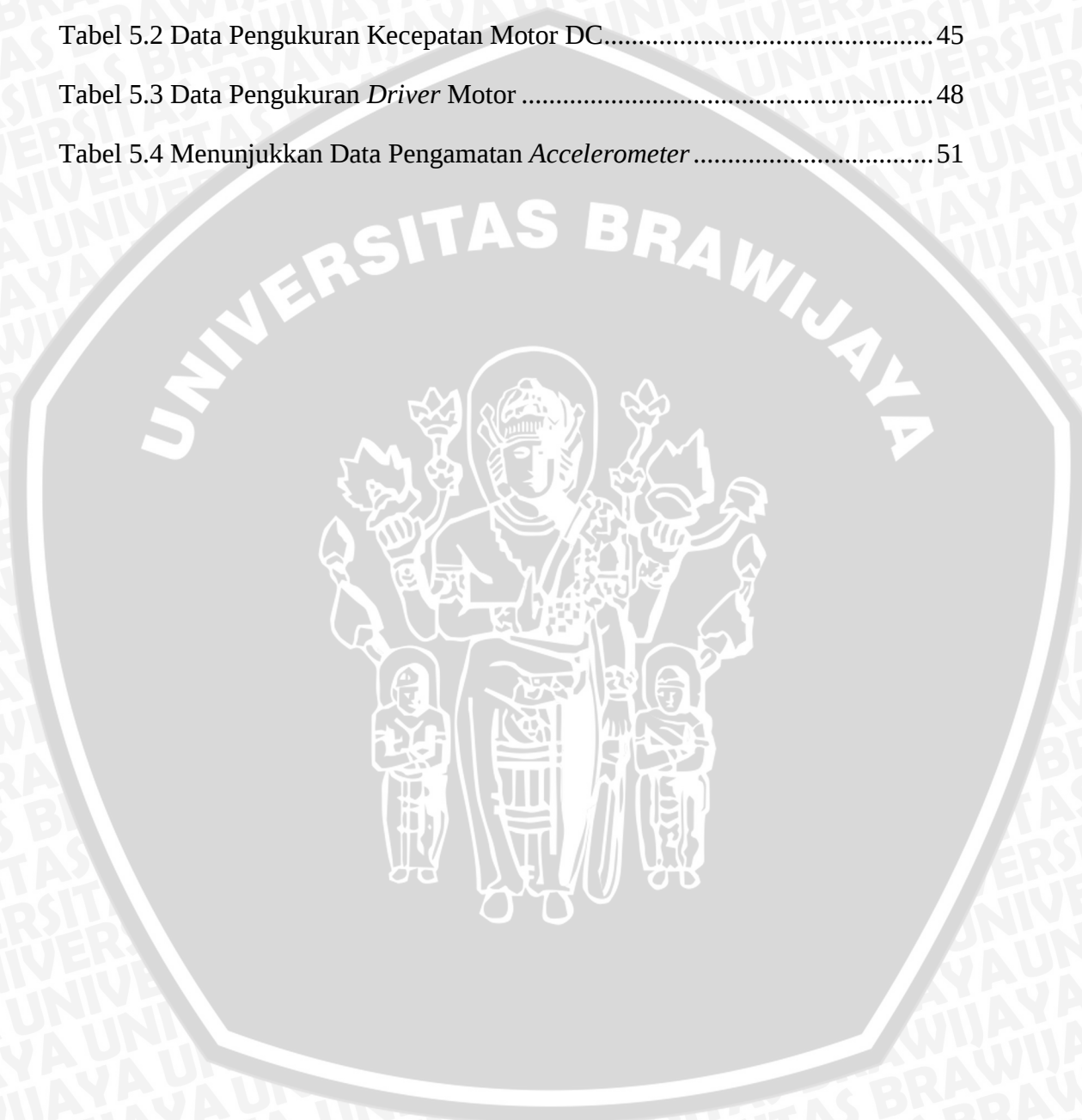
DAFTAR GAMBAR

Gambar 2.1 Prinsip Kerja Momen Inersia Pada Beberapa Benda	5
Gambar 2.2 Momen Inersia Pada Silinder Pejal	6
Gambar 2.3 Pengendali Fuzzy	7
Gambar 2.4 Struktur Dasar Logika Fuzzy	7
Gambar 2.5 Metode Penalaran Max-Min.....	10
Gambar 2.6 Inferensi Fuzzy dengan Metode MAX-DOT	11
Gambar 2.7 Analogi 3 Axis Accelerometer	12
Gambar 2.8 Gambar Melintang Motor DC	13
Gambar 2.9 Proses Kerja Motor DC	14
Gambar 2.10 Sinyal PWM Secara Umum	15
Gambar 2.11 Contoh Lebar <i>Duty Cycle Pulse Width Modulation</i> (PWM).....	16
Gambar 2.12 <i>Duty Cycle</i> untuk Perhitungan Tegangan <i>Output</i> Motor.....	16
Gambar 2.13 Pin Pada Relay DPDT	17
Gambar 2.14 Blok Diagram AT Mega 8 – 16 PU.....	19
Gambar 2.15 Konfigurasi Pin AT Mega 8	21
Gambar 4.1 Blok Diagram Sistem AKI	27
Gambar 4.2 Accelerometer MMA7455 3-Axis Accel Module.....	29
Gambar 4.3 Prinsip Kerja Accelerometer	29
Gambar 4.4 Skema Prinsip Dasar Kerja Relay DPDT.....	30
Gambar 4.5 Rangkaian Driver Motor DC.....	31
Gambar 4.6 Driver Motor DC	31
Gambar 4.7 Mikrokontroler ATMega8.....	32
Gambar 4.8 Modul Mikrokontroler ATMega8	33
Gambar 4.9 Implementasi Desain Bagian Pertama “Posisi <i>Disc Inerti</i> adan Posisi Sensor Accelerometer”	34

Gambar 4.10 Miniatur Sepeda Motor dan Mekanis Penyeimbang	34
Gambar 4.11 Motor DC gearbox 12 VDC 200rpm	35
Gambar 4.12 Fungsi Keanggotaan Kemiringan	37
Gambar 4.13 Fungsi Keanggotaan Perubahan Kemiringan	38
Gambar 4.14 Fungsi Keanggotaan Keluaran Motor DC	39
Gambar 4.15 Flowchart program	42
Gambar 5.1 Diagram Blok Pengujian Motor	44
Gambar 5.2 Grafik Pengukuran Kecepatan Motor DC	46
Gambar 5.3 Diagram Blok Pengujian <i>Driver</i> Motor	47
Gambar 5.4 Grafik Hubungan <i>Duty Cycle PWM</i> dan Tegangan Motor DC	49
Gambar 5.5 Grafik Hubungan <i>Duty Cycle PWM</i> dan Kecepatan Motor DC	49
Gambar 5.6 Diagram Blok Pengujian Accelerometer	50
Gambar 5.7 Grafik Pengujian <i>Accelerometer</i>	52
Gambar 5.8 Titik Keseimbangan Miniatur Sepeda Motor	53
Gambar 5.9 Blok Diagram Pengujian Sistem Secara Keseluruhan	54
Gambar 5.10 Pengambilan Data Pada Saat Posisi Miniatur Sepeda Motor Tidak Tegak Lurus atau 87° dari Permukaan Pijak	55
Gambar 5.11 Pengambilan Data Pada Saat Posisi Miniatur Sepeda Motor Tidak Tegak Lurus atau 89° dari Permukaan Pijak	56
Gambar 5.12 Pengambilan Data Pada Saat Posisi Miniatur Sepeda Motor Tidak Tegak Lurus atau 92° dari Permukaan Pijak	57
Gambar 5.13 Pengambilan Data Pada Saat Posisi Miniatur Sepeda Motor Tidak Tegak Lurus atau 139° dari Permukaan Pijak	58

DAFTAR TABEL

Tabel 4.1 Tabel Kaidah Atur Kemiringan.....	40
Tabel 5.1 Data Pengukuran Waktu Motor DC Mencapai Kecepatan Maksimum	45
Tabel 5.2 Data Pengukuran Kecepatan Motor DC.....	45
Tabel 5.3 Data Pengukuran <i>Driver</i> Motor	48
Tabel 5.4 Menunjukkan Data Pengamatan <i>Accelerometer</i>	51



ABSTRAK

Adhitya Yoga Guritna, Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya, Desember 2012, *Pengaturan Arah Dan Kecepatan Putar Disk Inersia Untuk Menjaga Keseimbangan Miniatur Sepeda Motor Menggunakan Kontrol Logika Fuzzy*, Dosen Pembimbing: Tri Nurwati. ST., MT. dan M. Aziz Muslim. ST., MT., Ph.D

Seiring dengan perkembangan teknologi dan ilmu pengetahuan terutama kemajuan dalam bidang robotika, mesin maupun robot dirancang agar mempunyai kemampuan yang sama bahkan melebihi kapasitas manusia, salah satu kemampuan manusia yang tidak dimiliki robot adalah kemampuan menjaga keseimbangan secara mandiri, di mana manusia memiliki sensor dan aktuator penyeimbang yang sangat kompleks. Miniatur sepeda motor dibuat berdasarkan betuk dari sepeda motor dengan *disk inersia* sebagai penyeimbang. Pada skripsi ini yang menjadi fokus bahasan adalah perubahan kecepatan dan arah dari *disk inersia* untuk menyeimbangkan miniatur sepeda motor.

Kontrol logika *fuzzy* merupakan salah satu metode pengontrolan yang sangat baik dan mudah dipahami, kontroler *fuzzy* diprogram pada sebuah mikrokontroler AVR AT Mega 8, yang berfungsi sebagai media penyimpanan dan akusisi data dari alat, sensor yang digunakan adalah *accelerometer*. Pemrograman dilakukan di *software* CVAVR dan untuk *flash* ke dalam mikrokontroler menggunakan *My Write*. Dalam perancangan *Fuzzy Logic Control (FLC)* digunakan metode mamdani, dengan inferensi max-min. dan dua masukan berupa kemiringan dan perubahan kemiringan. Dari parameter-parameter itu sistem dapat berjalan dengan baik dan mampu mempertahankan keseimbangan selama 10 detik.

Kata Kunci : Keseimbangan, perubahan arah dan kecepatan *disk inersia*, *Fuzzy Logic Control (FLC)*.

BAB I PENDAHULUAN

1.1 Latar Belakang

Seiring dengan perkembangan teknologi dan ilmu pengetahuan terutama kemajuan dalam bidang robotika, mesin maupun robot dirancang agar mempunyai kemampuan yang sama bahkan melebihi kapasitas manusia, salah satu kemampuan manusia yang tidak dimiliki robot adalah kemampuan menjaga keseimbangan secara mandiri, di mana manusia memiliki sensor dan aktuator penyeimbang yang sangat kompleks.

Dalam kehidupan sehari-hari kendaraan roda dua menjadi kendaraan yang paling banyak dipilih masyarakat sebagai kendaraan pribadi. Dimana kendaraan roda dua tidak dapat berdiri tegak tanpa adanya kontroler. Dalam skripsi sebelumnya (Putra, ARW, 2011) menyebutkan bahwa *“Keseimbangan pendulum terbalik dalam bergerak merupakan salah satu aplikasi bidang robotika yang sedang dikembangkan. Terutama pada robot berkaki dan beroda dua, Sehingga diperoleh mobilitas yang lebih tinggi.”* Dari hal itu munculah ide untuk membuat sistem keseimbangan otomatis pada miniatur sepeda roda dua, di mana sistem ini adalah sebuah sistem sensorik dan mekanis yang akan menjaga miniatur sepeda roda dua dalam keadaan seimbang dengan menggunakan sensor keseimbangan *“accelerometer”* dan aktuator berupa *“disk inertia”* dan kontroler yang ditanam ke dalam mikrokontroler Atmega8.

Kondisi miniatur sepeda roda dua tidaklah setimbang walaupun beberapa detik saja, agar tetap seimbang seluruh sensor dan aktuator harus bekerja secara sempurna. Yakni kecepatan putar dan arah penyeimbang harus sesuai dengan data yang diperoleh dari sensor keseimbangan, sehingga tiga elemen (sensor, aktuator, dan kontroler) dituntut bekerja secara sempurna.

Keseimbangan miniatur sepeda motor dijaga dengan menggunakan sensor keseimbangan *“accelerometer”*, aktuator penyeimbang *“disc inertia”* dan kontroler yang ditanam ke dalam mikrokontroler ATmega8. Karena pada sistem keseimbangan ini dibutuhkan respon dengan akurasi tinggi, maka

mekanik penyeimbang harus dicari titik keseimbangannya agar dapat dikalkulasi perkiraan sinyal kontrol yang dikeluarkan oleh kontroler *fuzzy* yang ditanam di mikrokontroler ATmega8.

1.2 Rumusan Masalah

Dari latar belakang masalah di atas, dibuat rumusan masalah sebagai berikut :

1. Bagaimana merancang *hardware* sebuah miniatur sepeda motor roda dua, beserta piringan inersia dan kontroler?
2. Bagaimana merancang *software* sebagai pembaca keseimbangan menggunakan *accelerometer*, dan kontroler *fuzzy* yang akan digunakan untuk mengendalikan arah dan kecepatan putar “*disk inertia*”?

1.3 Batasan Masalah

Dalam perancangan, permasalahan dibatasi oleh hal-hal sebagai berikut :

1. *Plant* yang digunakan adalah *miniatur sepeda motor menggunakan disk inertia* dengan spesifikasi sebagai berikut :
 - a. Motor yang digunakan adalah jenis motor DC *gearbox* 12VDC – 200rpm.
 - b. Rangka sepeda motor mempunyai panjang 160mm dan kedua rodanya berdiameter 55mm.
 - c. *Disk inertia* menggunakan *acrylic* dengan diameter 8,5mm dan tebal 3mm.
 - d. Sensor yang digunakan adalah *acceleometer MMA7455 3-Axis Accel Module*.
2. Metode kontrol yang digunakan adalah kontrol logika *fuzzy*.
3. Pada skripsi ini digunakan *software CodeVisionAVR Pro v2.03.9*.
4. Pusat pengontrolan adalah pada mikrokontroler ATmega8.
5. *Driver* motor yang digunakan adalah PWM mosfet IRFZ44 dan relay *Dual Pole Dual Throw (DPDT)*.

6. Miniatur sepeda motor hanya berdiri diam dan akan diberi gangguan berupa kemiringan sudut.

1.4 Tujuan

Tujuan yang ingin didapat pada penelitian skripsi ini adalah membuat model miniatur sepeda motor yang seimbang dengan adanya pengontrolan *disc inertia* tanpa adanya gangguan maupun diberi gangguan.

1.5 Manfaat

Desain dan implementasi sistem ini diharapkan dapat bermanfaat sebagai referensi untuk pengendalian sistem keseimbangan otomatis dan bermanfaat dalam pengembangan penelitian mengenai *balancing robot* dimasa mendatang.

1.6 Sistematika Pembahasan

BAB I

Pendahuluan

Bab ini berisi tentang uraian latar belakang, tujuan, batasan masalah, rumusan masalah, manfaat serta sistematika penulisan.

BAB II

Tinjauan Pustaka

Tinjauan Pustaka berisi dasar teori penunjang penelitian. Pustaka yang diambil adalah pustaka yang relevan dan sesuai serta mendukung penelitian, seperti buku-buku ilmu pengaturan, identifikasi, dan lain-lain. Selain dari buku pustaka juga akan diambil dari jurnal, internet, dan sumber pengetahuan yang lain.

BAB III

Metodologi Penelitian

Bab ini berisi tentang metode yang digunakan dalam pengerjaan alat seperti perancangan dan pembuatan rangkaian *interface*, pengujian alat, pengambilan data dan analisis data yang digunakan dalam skripsi ini.

BAB IV Perancangan dan Pembuatan Alat

Bab ini berisi tentang perancangan keseluruhan sistem pengaturan arah dan kecepatan putar disk inertia selanjutnya akan disingkat (AKI) pada penulisan selanjutnya. Setelah itu, bagaimana menerapkannya kontroler *fuzzy* ke dalam mikroprosesor AVR AT Mega 8 dan *software* di komputer, sehingga sistem dapat bekerja dengan baik.

BAB V Pengujian Alat

Memuat hasil pengujian terhadap alat yang telah direalisasikan.

BAB VI Kesimpulan dan Saran

Dalam bab ini, maka semua hal yang sudah dikerjakan pada bab sebelumnya, dianalisis, dan diambil kesimpulan. Serta rekomendasi dan saran untuk pengembangan alat.



BAB II

TINJAUAN PUSTAKA

2.1 Momen Inersia

Momen inersia adalah suatu besaran yang memperlihatkan tentang usaha suatu sistem benda untuk menentang gerak rotasinya. Besaran ini dimiliki oleh sistem benda (khususnya padat) apapun bentuknya (bulat, persegi, segitiga, dll). Oleh karena itu momen inersia didefinisikan sebagai kecenderungan suatu sistem benda untuk berputar terus atau diam sebagai reaksi terhadap gaya torsi dari luar.

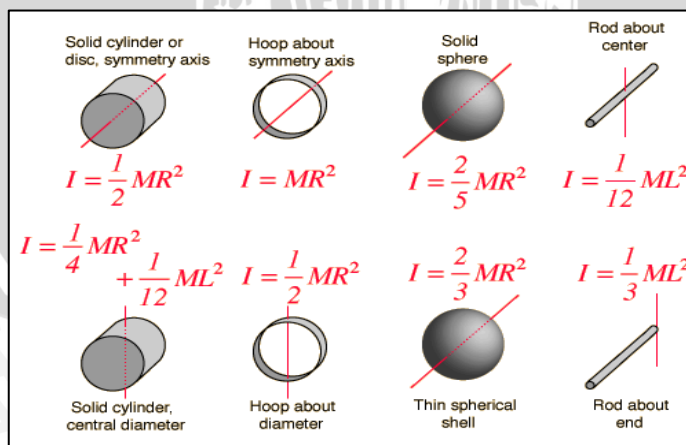
Pada dasarnya menentukan momen inersia benda berwujud tertentu seperti silinder pejal, bola dan sebagainya cenderung lebih mudah dibandingkan jika menentukan besar momen inersia untuk benda yang tidak beraturan dengan distribusi masa yang tidak sama. (Fisika, jilid 2, Halliday & Resnick).

Momen inersia dari beberapa partikel diperoleh dengan menjumlahkan secara aljabar biasa tiap-tiap momen inersia.

$$I = \sum_i m_i r_i^2 \quad (2-1)$$

Dimana I adalah momen inersia, m adalah massa benda dan r adalah jari-jari objek dari pusat massa (dalam beberapa kasus yang digunakan adalah panjang objek).

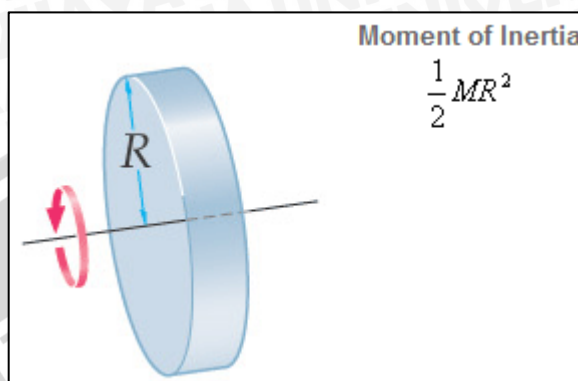
Momen inersia pada beberapa benda dapat dilihat pada Gambar 2.1.



Gambar 2.1 Prinsip Kerja Momen Inersia Pada Beberapa Benda

(<http://hyperphysics.phy-astr.gsu.edu/hbase/mi.html>)

Pada penelitian ini digunakan prinsip kerja momen inersia pada silinder pejal. *Disc inertia* yang digunakan merupakan silinder pejal, seperti terlihat pada Gambar 2.2.



Gambar 2.2 Momen Inersia Pada Silinder Pejal
(<http://www.livephysics.com/physical-constants/mechanics-pc/moment-inertia-uniform-objects/>)

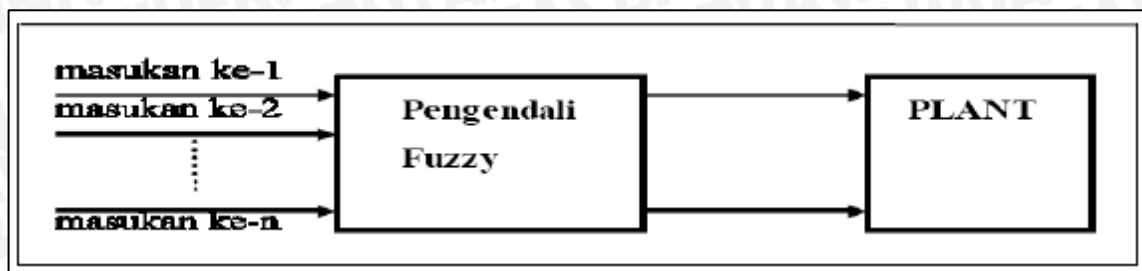
2.2 Kontrol Logika Fuzzy

Kontrol logika *fuzzy* adalah sistem berbasis aturan (*rule based system*) yang didalamnya terdapat himpunan aturan *fuzzy* yang mempresentasikan mekanisme pengambilan keputusan. Aturan yang dibuat digunakan untuk memetakan variabel *input* ke variabel *output* dengan pernyataan *If - Then*. Kontroler ini akan menggunakan data tertentu (*crisp*) dari sejumlah sensor kemudian mengubahnya menjadi bentuk linguistik atau fungsi keanggotaan melalui proses fuzzifikasi. Lalu dengan aturan *fuzzy*, *inference engine* yang akan menentukan hasil keluaran *fuzzy*. Setelah itu hasil ini akan diubah kembali menjadi bentuk numerik melalui proses defuzzifikasi. Pengendali *fuzzy* secara umum merupakan pengendali kalang tertutup.

Struktur dasar pengendali logika *fuzzy* terdiri dari unit fuzzifikasi, mekanisme penentuan keputusan dan basis aturan (*inference system*), dan unit defuzzifikasi (dwiasta, 2002, 3).

2.2.1 Struktur Dasar Kontrol Logika Fuzzy

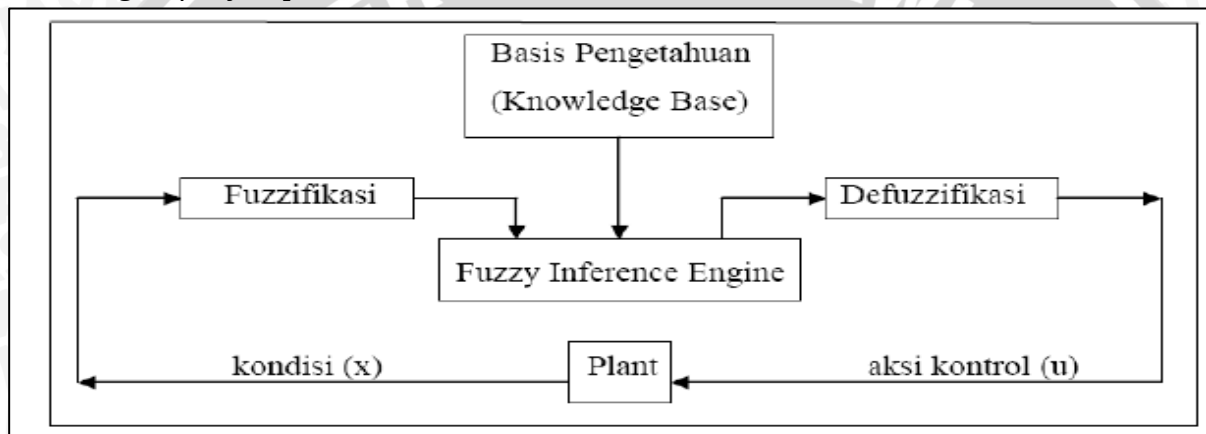
Dalam sistem pengendalian dengan logika *fuzzy* dilibatkan suatu blok pengendali yang menerima satu atau lebih masukan dan mengumpankan satu atau lebih keluaran ke *plant* atau blok lain sebagaimana terlihat dalam Gambar 2.3



Gambar 2.3 Pengendali Fuzzy

Sumber : Donald Coughanowr, 1991

Komponen utama penyusun kontrol logika *fuzzy* adalah unit fuzzifikasi, *fuzzy inference*, basis pengetahuan dan unit defuzzifikasi. Struktur dasar kontrol logika *fuzzy* dapat dilihat dalam Gambar 2.4.



Gambar 2.4 Struktur Dasar Logika Fuzzy

Sumber : Jun Yan, 1994.

Basis pengetahuan terdiri dari dua jenis :

- **Basis data**

Mendefinisikan parameter *fuzzy* sebagai bagian dari himpunan *fuzzy* dengan menentukan batas-batas fungsi keanggotaan pada semesta pembicaraan untuk tiap-tiap variabel.

- **Basis aturan**

Memetakan nilai masukan *fuzzy* menjadi nilai keluaran *fuzzy*.

2.2.2 Fuzzifikasi

Proses fuzzifikasi merupakan proses untuk mengubah variabel non-*fuzzy* (variable numerik) menjadi variabel *fuzzy* (variabel linguistik). Nilai masukan-masukan yang masih dalam bentuk variabel numerik yang telah dikuantisasi

sebelum diolah oleh pengendali logika *fuzzy* harus diubah terlebih dahulu ke dalam variable *fuzzy*. Melalui fungsi keanggotaan yang telah disusun, maka dari nilai-nilai masukan tersebut menjadi informasi *fuzzy* yang berguna nantinya untuk proses pengolahan secara *fuzzy* pula. Proses ini disebut *fuzzifikasi*. (Jun Yan, 1994 : 49).

Proses *fuzzifikasi* diekspresikan sebagai berikut:

$$x = \text{fuzzifier}(x_0) \dots\dots\dots (2-2)$$

dengan :

x_0 = nilai *crisp* variabel masukan

x = himpunan *fuzzy* variabel yang terdefinisi

fuzzifier = operator *fuzzifikasi* yang memetakan himpunan *crisp* ke himpunan *fuzzy*

Pedoman memilih fungsi keanggotaan untuk proses *fuzzifikasi*, menurut Jun Yan, menggunakan :

1. Himpunan *fuzzy* dengan distribusi simetris.
2. Gunakan himpunan *fuzzy* dengan jumlah ganjil, berkaitan erat dengan jumlah kaidah (*rules*).
3. Mengatur himpunan *fuzzy* agar saling menumpuk.
4. Menggunakan fungsi keanggotaan bentuk segitiga atau trapesium.

2.2.3 Aturan Fuzzy (Fuzzy Rule)

Fuzzy rule adalah bagian yang menggambarkan dinamika suatu sistem terhadap masukan yang dikarakteristikan oleh sekumpulan variabel-variabel linguistik dan berbasis pengetahuan seorang operator ahli. Pernyataan tersebut umumnya dinyatakan oleh suatu pernyataan bersyarat. Dalam pengendali berbasis *fuzzy* aturan pengendalian *fuzzy* berbentuk aturan “ *IF – THEN*”. Untuk sebuah

sistem *MISO* (*Multi Input Single Output*) basis aturan pengendalian *fuzzy* berbentuk seperti berikut ini.

Rule 1 IF X1 is A11 AND ... AND xm is A1m THEN Y is B1

Rule 2 IF X1 is A21 AND ... AND xm is A2m THEN Y is B2

..

dan seterusnya

..

Rule n IF Xn is An1 AND ... AND xm is Anm THEN Y is Bn

Dengan X_j merupakan variabel masukan sistem, A_{ij} merupakan *fuzzy* set untuk X_j , Y merupakan variabel keluaran sistem, B_i merupakan *fuzzy* set untuk Y , *AND* adalah operator *fuzzy*.

2.2.4 Metode Inferensi

Metode inferensi merupakan proses untuk mendapatkan keluaran dari suatu kondisi masukan dengan mengikuti aturan-aturan yang telah ditetapkan. Keputusan yang didapatkan pada proses ini masih dalam bentuk *fuzzy* yaitu derajat keanggotaan keluaran. Ada beberapa cara untuk mengidentifikasi aturan mana yang akan dipakai dengan menggunakan nilai keanggotaan masukan. Menurut Jun Yan, diantara bermacam metode inferensi *fuzzy* ada dua metode yang paling sering digunakan pada kontrol logika *fuzzy* yaitu :

2.2.4.1 Metode INFERENSI MAX – MIN

Pada metode *Max – Min* aturan operasi *minimum* Mamdani digunakan untuk implikasi *fuzzy*. Persamaan aturan *minimum* adalah

$$\bigcup_{i=1}^n \alpha_i \wedge \mu_{ci} \dots \dots \dots (2-3)$$

dengan

$$\alpha_i = \mu_{A_i}(x_0) \wedge \mu_{B_i}(y_0) \dots\dots\dots (2-4)$$

Sebagai contoh , terdapat dua basis kaidah atur *fuzzy*, yaitu :

R1 : Jika x adalah A1 dan y adalah B1 maka z adalah C1

R2 : Jika x adalah A2 dan y adalah B2 maka z adalah C2

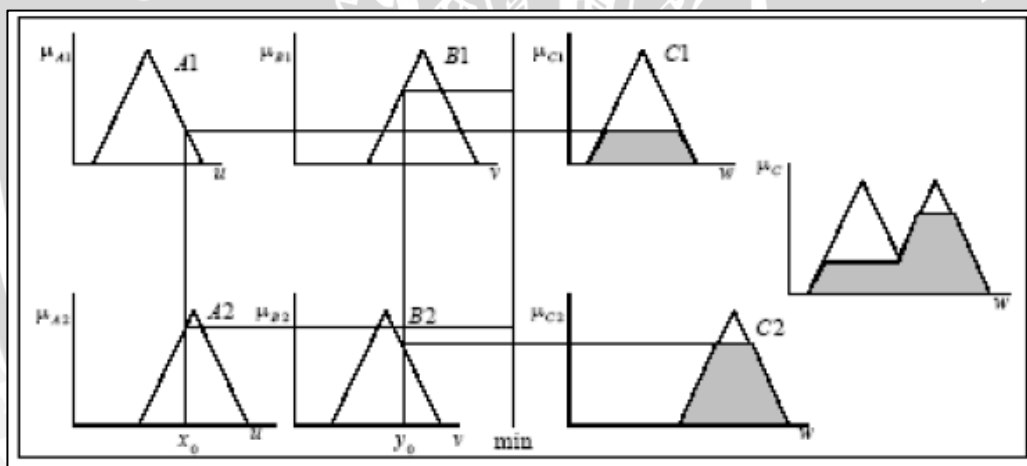
Pada metode penalaran *MAX-MIN* fungsi keanggotaan konsekuen dinyatakan dengan

Dimana

$$\alpha_1 = \mu_{A1}(x_0) \wedge \mu_{B1}(y_0) \dots\dots\dots (2-5)$$

$$\alpha_2 = \mu_{A2}(x_0) \wedge \mu_{B2}(y_0) \dots\dots\dots (2-6)$$

Lebih jelas metode ini dideskripsikan dalam Gambar 2.5



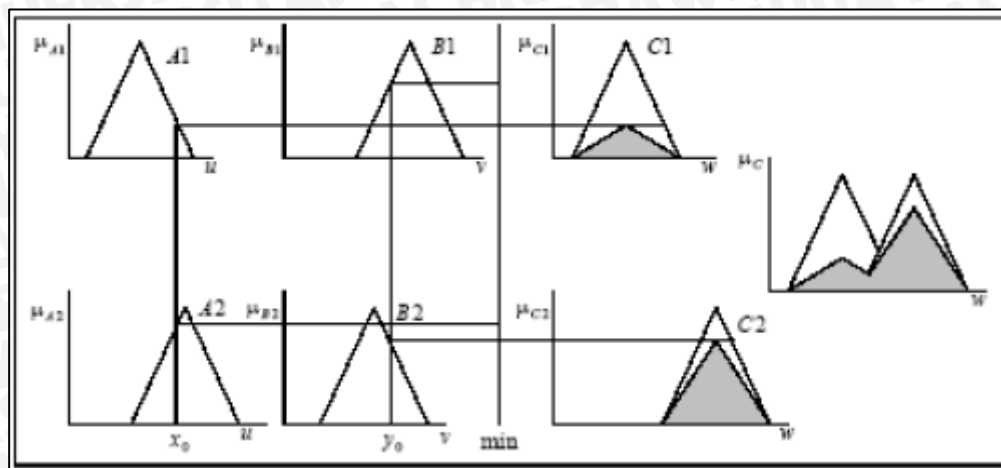
Gambar 2.5 Metode Penalaran Max-Min

2.2.4.2 Metode INFERENSI MAX – DOT

Pada metode inferensi *Max – Dot*, aturan operasi *Larsen* digunakan untuk fungsi implikasi *fuzzy*. Metode *MAX-DOT* fungsi keanggotaan konsekuen C dinyatakan dengan:

$$\mu_c(\omega) = (\alpha_1 \bullet \mu_{c1}(\omega)) \vee (\alpha_2 \bullet \mu_{c2}(\omega)) \dots\dots\dots (2-7)$$

Metode penalaran *MAX-DOT* diperlihatkan dalam Gambar 2.6.



Gambar 2.6 Inferensi Fuzzy dengan Metode MAX-DOT

Sumber : Jun Yan, 1994 : 5

2.2.5 Defuzzifikasi

Defuzzifikasi adalah proses untuk mendapatkan nilai numerik dari data fuzzy yang dihasilkan dari proses inferensi. (Jun Yan, 1994 : 55). Proses defuzzifikasi dinyatakan sebagai berikut :

$$y_0 = \text{defuzzifier}(y) \dots\dots\dots (2-8)$$

dengan:

- y : aksi kontrol *fuzzy*.
- y₀ : aksi kontrol *crisp*.
- Defuzzifier : operator defuzzifikasi

Dua metode defuzzifikasi yang umum digunakan adalah :

1. Metode COA (*Center Of Area*)

Metode ini didefinisikan sebagai berikut:

$$U = \frac{\sum_{i=1}^n w_i u_i}{\sum_{i=1}^n w_i} \dots\dots\dots (2-9)$$

dengan:

- U = *output*
- w_i = bobot nilai benar w_i

- ui = nilai linguistik pada fungsi keanggotaan *output*
 n = banyak derajat keanggotaan

2. Metode MOM (*Mean of Maximum*)

Metode ini didefinisikan sebagai berikut :

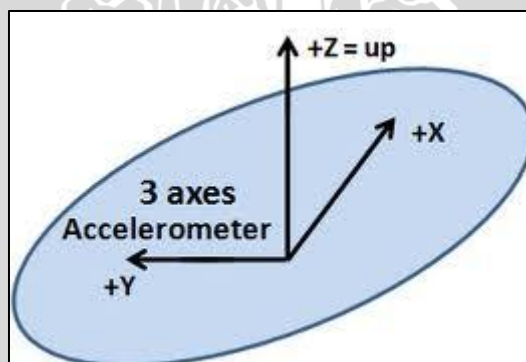
$$Z = \frac{\sum_{i=1}^n z_i}{n} \dots\dots\dots (2-10)$$

dengan :

- n = jumlah proposisi / *domain*
 Z = *output*
 Zi = *domain* yang memiliki derajat keanggotaan terbesar / maximum.

2.3 Accelerometer

Accelerometer adalah suatu transduser untuk mengukur percepatan sehingga dapat mendeteksi adanya perubahan posisi *device* dan berapa banyak perubahan itu terjadi. *Accelerometer* merupakan sensor yang peka terhadap gerak linier terhadap 3 sumbu gerakan yakni kanan-kiri, atas-bawah dan depan-belakang. Gambar 2.7 menunjukkan daerah kerja *accelerometer*.



Gambar 2.7 Analogi 3 Axis Accelerometer

(<http://memsblog.wordpress.com/2012/04/04/degrees-of-freedom-vs-axes/>)

Hampir semua *accelerometer* sudah dalam bentuk digital dan bukan dengan sistem mekanik sehingga cara kerjanya hanya berdasarkan temperatur yang diolah secara digital dalam satu chip.

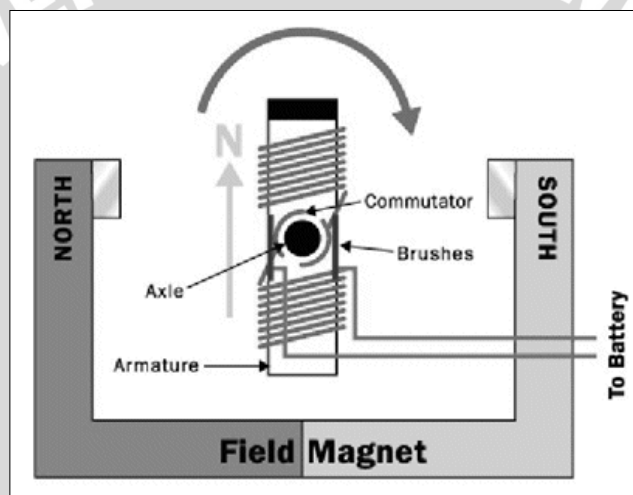
2.4 Motor DC

2.4.1 Prinsip Kerja Motor DC

Segulung kawat yang dialiri arus listrik dan ditempatkan di dalam suatu medan magnet akan mengalami gaya yang sebanding dengan arus dan kekuatan medan magnetnya. Gaya yang ditimbulkan disebut dengan Gaya Lorentz [Soemarwanto, 1997:28] yang dapat dirumuskan sebagai berikut:

$$F = B \cdot I \cdot L \dots (N) \dots\dots\dots (2-11)$$

Dalam hal ini B adalah kerapatan *fluks* magnet (Wb/m^2), I adalah arus yang mengalir (A) dan L adalah panjang kawat (m). Gambar melintang dari motor DC ditunjukkan pada gambar 2.8



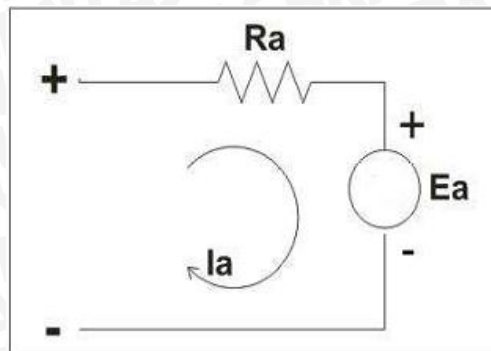
Gambar 2.8 Gambar Melintang Motor DC

Sumber: www.electronics-scheme.com

Persamaan 2.11 merupakan prinsip dasar sebuah motor, di mana terjadinya proses perubahan energi listrik menjadi energi mekanik. Jika motor mempunyai jari-jari sebesar r , maka akan menimbulkan torsi sebesar :

$$\tau = F \cdot r = B \cdot I \cdot L \cdot r \dots (Nm) \dots\dots\dots (2-12)$$

Pada saat dibangkitkan, konduktor akan bergerak di dalam medan magnet dan akan menimbulkan gaya gerak listrik (ggl) yang merupakan reaksi lawan terhadap tegangan penyebabnya. Proses konversi energi listrik menjadi energi mekanik dapat berlangsung jika tegangan sumber lebih besar dari gaya gerak listrik lawan.



Gambar 2.9 Proses Kerja Motor DC

Sumber: www.mikron123.com

Motor dapat berputar jika tegangan masukan motor lebih besar dari ggl yang timbul. Hubungan antara tegangan sumber dan ggl lawan seperti ditunjukkan dalam Gambar 2.9 dapat dirumuskan sebagai berikut:

$$Ea = Vin - Ia.Ra \dots(V) \quad \dots\dots\dots (2-13)$$

Dalam hal ini Ea adalah tegangan pada jangkar, Vin adalah tegangan masukan, Ia adalah arus jangkar dan Ra adalah tahanan jangkar, sedangkan induksi yang timbul adalah:

$$Ea = C n \Phi \dots(V) \quad \dots\dots\dots (2-14)$$

Dengan C adalah konstanta, n adalah kecepatan motor, dan Φ adalah fluks magnetik yang besarnya sebanding dengan arus penguatan torsi. Torsi pada motor juga sebanding dengan fluks magnetik dan arus. Hal ini ditunjukkan dalam Persamaan 2.15 berikut :

$$\tau = C \Phi Ia \dots(Nm) \quad \dots\dots\dots (2-15)$$

Jika diketahui kecepatan sudut ω adalah :

$$\omega = \frac{2\pi.n}{60} \dots(rad/s) \quad \dots\dots\dots (2-16)$$

Maka hubungan torsi dan kecepatan motor adalah :

$$\tau = P/\omega \dots (Nm) \quad \dots\dots\dots (2-17)$$

$$\tau = \frac{P}{2\pi n/60} \dots(Nm) \quad \dots\dots\dots (2-18)$$

2.4.2 Pengaturan Motor DC

Apabila persamaan 2-13 disubstitusikan dengan persamaan 2-14 akan didapatkan rumus kecepatan motor (n) sebagai berikut :

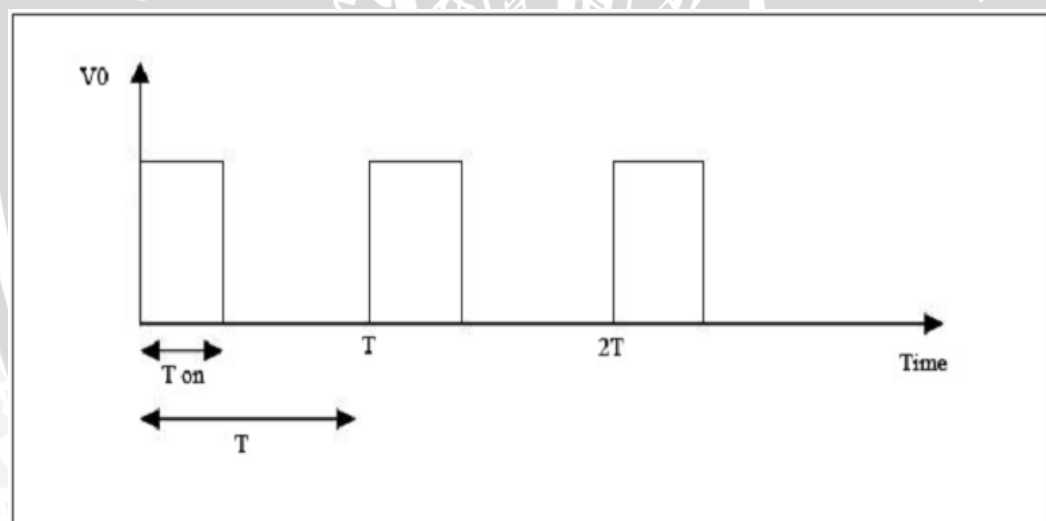
$$n = \frac{V_{in} - I_a R_a}{C \Phi} \dots (\text{rpm}) \quad \dots \dots \dots (2-19)$$

Dari Persamaan 2.19 terlihat bahwa kecepatan motor sebanding dengan tegangan masukan (V_{in}) [Zuhal, 1993: 91]. Jadi apabila tegangan masukan besar maka kecepatan motor akan cepat, demikian pula sebaliknya, jika tegangan masukan kecil maka kecepatan motor akan lambat.

2.5 PWM (Pulse Width Modulation)

PWM (*Pulse Width Modulation*) digunakan untuk mengatur kecepatan dari motor DC. Dimana kecepatan motor DC tergantung pada besarnya *duty cycle* yang diberikan pada motor DC tersebut.

Pada sinyal PWM, frekuensi sinyal konstan sedangkan *duty cycle* bervariasi dari 0%-100%. Dengan mengatur *duty cycle* akan diperoleh keluaran yang diinginkan. Sinyal PWM secara umum dapat dilihat dalam Gambar 2.10



Gambar 2.10 Sinyal PWM Secara Umum
Sumber: www.electronics-scheme.com

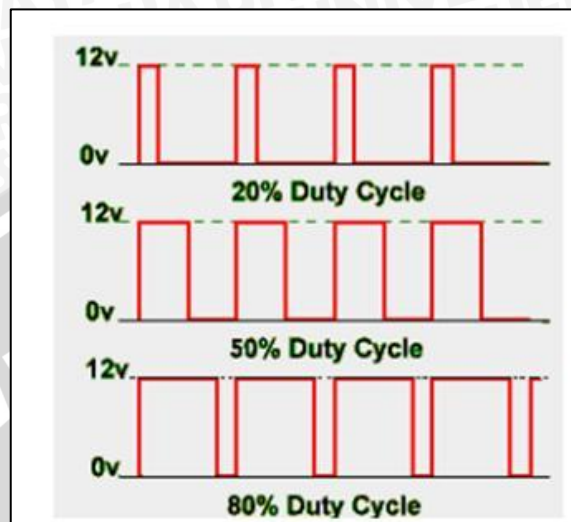
$$Duty cycle = \frac{T_{on}}{T} \times 100\% \dots (\%) \quad \dots \dots \dots (2-20)$$

Dimana :

T_{on} = Periode logika tinggi

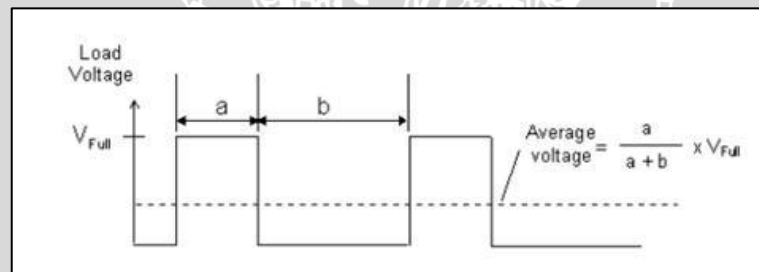
T = Periode keseluruhan

Duty cycle 100% berarti sinyal tegangan pengatur motor dilewatkan seluruhnya. Jika tegangan catu 100V, maka motor akan mendapat tegangan 100V. pada *duty cycle* 50%, tegangan pada motor hanya akan diberikan 50% dari total tegangan yang ada, begitu seterusnya. Contoh lebar *duty cycle* dapat dilihat pada Gambar 2.11.



Gambar 2.11 Contoh Lebar *Duty Cycle Pulse Width Modulation* (PWM)
(<http://digilib.ittelkom.ac.id>)

Perhitungan Pengontrolan tegangan *output* motor dengan metode PWM cukup sederhana.



Gambar 2.12 *Duty Cycle* untuk Perhitungan Tegangan *Output* Motor
(<http://digilib.ittelkom.ac.id>)

Dengan menghitung *duty cycle* yang diberikan, akan didapat tegangan *output* yang dihasilkan. Sesuai dengan rumus yang telah dijelaskan pada gambar 2.12.

$$\text{Average Voltage} = \frac{a}{a + b} \times V_{full} \dots\dots\dots (2-21)$$

Average voltage merupakan tegangan *output* pada motor yang dikontrol oleh sinyal PWM. *a* adalah nilai *duty cycle* saat kondisi sinyal “on”. *b* adalah nilai *duty cycle* saat kondisi sinyal “off”. *V full* adalah tegangan maximum pada

motor. Dengan menggunakan persamaan 2-21, maka akan didapatkan tegangan output sesuai dengan sinyal kontrol PWM yang dibangkitkan.

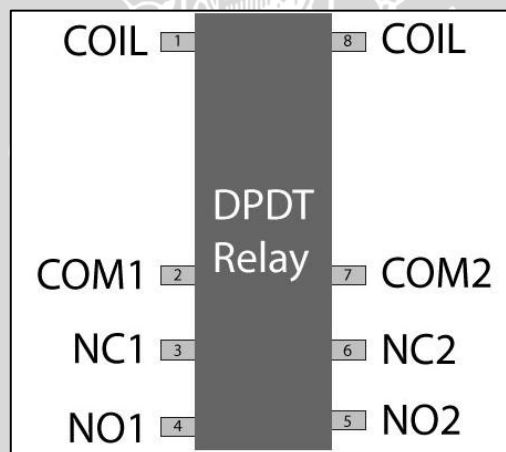
2.6 Driver Motor DC

Driver Motor DC yang digunakan adalah modul relay *dual pole dual throw* (DPDT) sebagai pengatur arah putaran motor DC dan rangkaian mosfet yang berfungsi sebagai penguat tegangan masukan dari mikrokontroler ATmega8.

Relay adalah sebuah saklar elektronis yang dapat dikendalikan dari rangkaian elektronik lainnya. Relay terdiri dari 3 bagian utama, yaitu:

1. koil : lilitan dari relay
2. *common* : bagian yang tersambung dengan *Normally Open* (NO)
3. kontak : terdiri dari *Normally Close* (NC) dan *Normally Open* (NO)

Gambar skematik pin dari relay DPDT ditunjukkan pada Gambar 2.13.



Gambar 2.13 Pin Pada Relay DPDT

Relay DPDT dapat digunakan untuk mengganti atau membalik polaritas dari terminal *device* yang terhubung dengan *output*. Sebagai contohnya yaitu untuk mengontrol motor DC searah jarum jam dan berlawanan arah jarum jam, dapat menggunakan kombinasi berikut. Pin 2 dan 7 masing-masing dihubungkan ke Vcc dan *ground*. Terminal motor awalnya dihubungkan ke pin 3 dan 4 sementara terminal lainnya terhubung ke pin 5 dan 6. Pada saat tidak ada sinyal *input* yang diberikan maka motor akan berputar dalam satu arah putaran (misalnya, searah jarum jam), dan pada saat diberikan sinyal *input*, posisi kontak

akan berubah dan menyebabkan perubahan arah putaran motor (berlawanan arah jarum jam)

2.7 AVR AT Mega 8

AVR merupakan seri mikrokontroler CMOS 8-bit buatan Atmel, berbasis arsitektur RISC (*Reduced Instruction Set Computer*). Hampir semua instruksi dieksekusi dalam satu siklus *clock*. AVR mempunyai 32 register *general-purpose*, *timer/counter* fleksibel dengan *mode compare*, *interrupt* internal dan eksternal, serial USART, *programmable Watchdog Timer*, dan *mode power saving*. Mempunyai PWM internal. AVR juga mempunyai *In-System Programmable Flash on-chip* yang mengijinkan memori program untuk diprogram ulang dalam sistem menggunakan hubungan serial SPI.

AT Mega 8 merupakan satu jenis mikrokontroler AVR dengan 18 jalur *input/output* yang dapat diprogram dalam kemasan 20 pin, mikrokontroler *AT Mega 8* memiliki keistimewaan mikrokontroler jenis AVR, Beberapa keistimewaan dan fitur-fitur dari mikrokontroler AVR *AT Mega 8* antara lain :

Special Microcontroller Features

- *Debug WIRE On-chip Debugging*
- *In-System Programmable via SPI Port*
- *External and Internal Interrupt Sources*
- *Low-power Idle, Power-down, and Standby Modes*
- *Enhanced Power-on Reset Circuit*
- *Programmable Brown-out Detection Circuit*
- *Internal Calibrated Oscillator*

I/O and Packages

- *18 Programmable I/O Lines*
- *20-pin PDIP, 20-pin SOIC, 20-pad QFN/MLF*

Operating Voltages

- 1.8 – 5.5V (*AT Mega 8V*)
- 2.7 – 5.5V (*AT Mega 8*)

Speed Grades

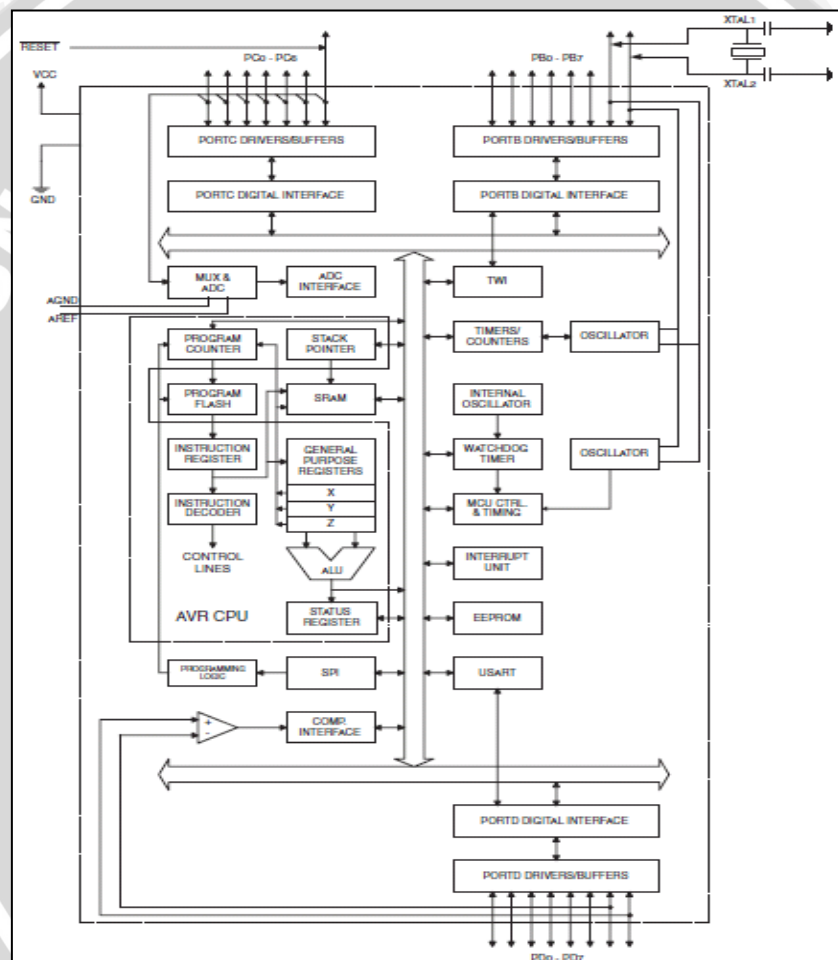
- AT Mega 8V: 0 – 4 MHz @ 1.8 – 5.5V, 0 – 10 MHz @ 2.7 -5.5V
- AT Mega 8: 0 – 10 MHz @ 2.7 – 5.5V, 0 – 20 MHz @ 4.5 – 5.5V

Typical Power Consumption

- Active Mode 1 MHz, 1.8V: 230 μ A 32 kHz, 1.8V: 20 μ A (including oscillator)
- Power-down Mode < 0.1 μ A at 1.8V

2.7.1 Arsitektur Mikrokontroler AT Mega 8

Berikut menunjukkan diagram blok dari mikrokontroler AT Mega 8.



Gambar 2.14 Blok Diagram AT Mega 8 – 16 PU

Sumber : datasheet ATMEL AT Mega 8 and 8L

Dari Gambar 2.14 dapat di lihat bahwa blok diagram AT Mega 8 memiliki beberapa bagian diantaranya :

1. Saluran I/O sebanyak 23 buah yaitu, port B, port C, dan port D.

2. ADC sebanyak 6 saluran, kecuali PC 5 dan PC 6 yang memiliki fungsi tambahan sebagai SCL dan SDA.
3. Sebuah port komunikasi serial USART.
4. *Watchdog Timer* dengan osilator internal.
5. *External and internal interrupts*
6. Tiga buah *timers/counters* dengan mode pembandingan, yang terhubung dengan osilator.
7. CPU yang terdiri dari 32 buah register yang bekerja.
8. SRAM sebesar 1 *Kilo byte*.
9. 512 bytes EEPROM yang dapat di program pada saat dioperasikan.
10. Antarmuka komputer analog.

2.7.2 Fitur AT Mega 8

Kapabilitas *detail* dari AT Mega 8 adalah sebagai berikut :

1. Sistem Mikroprosesor 8 bit berbasis RISC dengan kecepatan maksimal 16 MHz.
2. Kapasitas memori *flash* sebesar 8 Kbytes, 512 byte EEPROM, internal SRAM 512 bytes.
3. ADC internal sebanyak 6 *channel*.
4. Port komunikasi serial USART.
5. Penguncian terprogram untuk keamanan *software*.

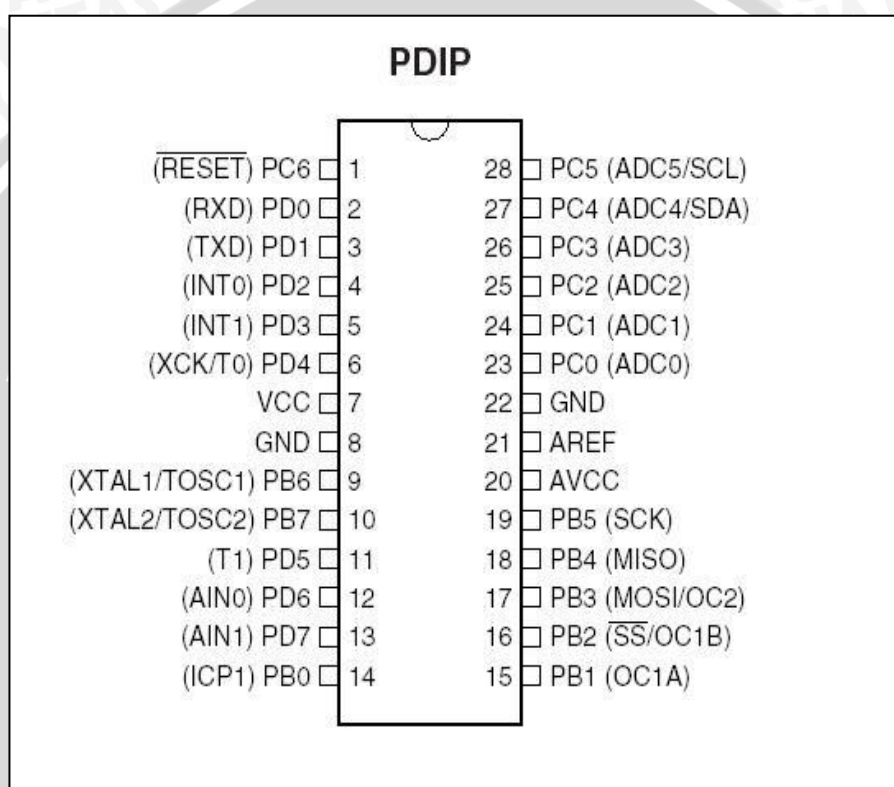
2.7.3 Konfigurasi AT Mega 8

Penjelasan dari fungsi pin-pin yang terdapat pada AT Mega 8 adalah sebagai berikut :

1. VCC sebagai pin masukkan catu daya
2. GND sebagai pin *ground*
3. AREF Analog *input* referensi untuk ADC
4. AVCC Catu daya untuk ADC
5. XTAL 1 *input* untuk *inverting oscillator amplifier* dan *input* bagi *clock* internal
6. XTAL 2 *output inverting oscillator amplifier*
7. Port B merupakan port I/O 8 bit dua arah yang memiliki resistor *pull-up* didalamnya.

8. Port C merupakan port I/O 7 bit dua arah dengan resistor *pull-up* didalamnya yang juga berfungsi sebagai keluaran ADC.
9. Port D merupakan port I/O 8 bit dua arah dengan resistor *pull-up* didalamnya yang berfungsi sebagai port *input* analog.

Gambar 2.15 merupakan penjelasan dari pin-pin yang terdapat pada AT Mega 8



Gambar 2.15 Konfigurasi Pin AT Mega 8
Sumber: Datasheet ATMEL AT Mega 8 dan 8L

2.8 Program Code Vision AVR

CodeVisionAVR merupakan sebuah cross-compiler C, *Integrated Development Environment* (IDE), dan *Automatic Program Generator* yang didesain untuk mikrokontroler buatan Atmel seri AVR. CodeVisionAVR dapat dijalankan pada sistem operasi Windows 95, 98, Me, NT4, 2000, dan XP.

Cross-compiler C mampu menerjemahkan hampir semua perintah dari bahasa ANSI C, sejauh yang diijinkan oleh arsitektur dari AVR, dengan tambahan beberapa fitur untuk mengambil kelebihan khusus dari arsitektur AVR dan kebutuhan pada sistem *embedded*.

File object COFF hasil kompilasi dapat digunakan untuk keperluan *debugging* padatingkatan C, dengan pengamatan variabel, menggunakan *debugger* Atmel AVR Studio.

IDE mempunyai fasilitas internal berupa software AVR *Chip In-System Programmer* yang memungkinkan pengguna untuk melakukan *transfer* program kedalam chip mikrokontroler setelah sukses melakukan kompilasi/assembly secara otomatis. *Software In-System Programmer* didesain untuk bekerja dengan Atmel STK500/AVRISP/AVRProg, Kanda Systems STK200+/300, Dontronics DT006, Vogel Elektronik VTEC-ISP, Futurlec JRAVR dan MicroTronics ATCPU/Mega2000 *programmers/development boards*.

Untuk keperluan *debugging* sistem *embedded*, yang menggunakan komunikasi serial, IDE mempunyai fasilitas internal berupa sebuah Terminal. Selain *library* standar C, CodeVisionAVR juga mempunyai *library* tertentu untuk:

- a) Modul LCD *alphanumeric*
- b) Bus I2C dari Philips
- c) Sensor Suhu LM75 dari National Semiconductor
- d) *Real-Time Clock*: PCF8563, PCF8583 dari Philips, DS1302 dan DS1307 dari Maxim/Dallas Semiconductor
- e) Protokol 1-Wire dari Maxim/Dallas Semiconductor
- f) Sensor Suhu DS1820, DS18S20, dan DS18B20 dari Maxim/Dallas Semiconductor
- g) Termometer/Termostat DS1621 dari Maxim/Dallas Semiconductor
- h) EEPROM DS2430 dan DS2433 dari Maxim/Dallas Semiconductor
- i) SPI
- j) *Power Management*
- k) *Delay*
- l) Konversi ke Kode Gray

CodeVisionAVR juga mempunyai *Automatic Program Generator* bernama CodeWizardAVR, yang mengijinkan pengguna untuk menulis, dalam hitungan menit, semua instruksi yang diperlukan untuk membuat fungsi-fungsi berikut:

- a) *Set-up* akses memori eksternal
- b) Identifikasi sumber *reset* untuk *chip*
- c) Inisialisasi port *input/output*
- d) Inisialisasi interupsi eksternal
- e) Inisialisasi *Timer/Counter*
- f) Inisialisasi *Watchdog-Timer*
- g) Inisialisasi UART (USART) dan komunikasi *serial* berbasis *buffer* yang digerakkan oleh interupsi
- h) Inisialisasi Pembanding Analog
- i) Inisialisasi ADC
- j) Inisialisasi Antarmuka SPI
- k) Inisialisasi Antarmuka Two-Wire
- l) Inisialisasi Antarmuka CAN
- m) Inisialisasi Bus I2C, Sensor Suhu LM75, *Thermometer/Thermostat* DS1621 dan *Real-Time Clock* PCF8563, PCF8583, DS1302, dan DS1307
- n) Inisialisasi *Bus 1-Wire* dan Sensor Suhu DS1820, DS18S20
- o) Inisialisasi model LCD

BAB III

METODOLOGI

Penelitian dalam kajian skripsi ini merupakan penelitian yang bersifat aplikatif, yaitu merencanakan dan membuat *AKI* dengan menggunakan kontrol logika fuzzy sebagai kendali, tentu saja dengan tujuan agar dapat memiliki performansi yang responsif dan sesuai dengan sistem yang sudah direncanakan dan dirancang.

Agar dapat direalisasikan pembuatan rancangan tersebut, terdapat langkah-langkah yang bisa dilakukan, diantaranya sebagai berikut :

3.1 Studi literatur

Studi literatur mengacu pada prinsip kerja alat yang dirancang meliputi sistem pengendalian *arah dan kecepatan putar disk inertia* dengan menggunakan FLC (*Fuzzy Logic Controller*), karakteristik dari komponen yang digunakan serta studi tentang media pengendali dan akuisisi data yaitu menggunakan AVR *ATmega 8* produksi dari ATMEL sebagai pusat pengontrol FLC.

Adapun hal-hal yang berhubungan dengan hal tersebut adalah:

a. Studi tentang proses pengendalian *AKI*

- Karakteristik dari sistem *AKI*

b. Studi tentang rangkaian sensor

- Karakteristik spesifikasi *accelerometer*.
- Penerapan *accelerometer* sesuai sistem.

c. Kontrol Logika Fuzzy

- Karakteristik umum kontroler *fuzzy*, beserta sistematika perancangan kontrol logika *fuzzy*.
- Teknik pemrograman AVR *ATmega 8* sebagai pusat akuisisi data dari kontroler *fuzzy*.

3.2 Perancangan Sistem

Perancangan sistem dilakukan sebagai langkah awal sebelum terbentuknya suatu sistem beserta rangkaian elektronik pendukungnya, hal ini dimaksudkan

agar sistem *AKI* tersebut dapat berjalan sesuai dengan deskripsi awal yang telah direncanakan. Perancangan sistem yang dilakukan meliputi:

- Penentuan dimensi dan bahan yang digunakan dalam *AKI*.
- Penentuan rangkaian elektronik yang digunakan.
- Penentuan spesifikasi sistem yang akan dibuat, meliputi :
 - a. Penentuan deskripsi kerja sistem secara keseluruhan.
 - b. Kontroler *fuzzy* dan rangkaian elektronik pendukung.
- Perancangan program pada *AVR AT Mega 8* sebagai media kendali dan akuisisi data kontroler *fuzzy*.

3.3 Realisasi Pembuatan Sistem

Untuk merealisasikan alat atau sistem yang telah dirancang sebelumnya maka perlu diperhatikan beberapa tahapan dalam pembuatan alat tersebut, diantaranya yaitu :

- Pembuatan mekanik *AKI* secara keseluruhan yang meliputi penentuan batasan sudut dan pemasangan serta pemilihan motor, penentuan material dari kerangka miniatur sepeda motor dan *disk inertia*.
- Pembuatan rangkaian elektronik meliputi rangkaian *acceleometer* dan penguat tegangan, dan sistem minimal dari *AT Mega 8*.
- Pengisian program yang telah dirancang sebelumnya pada *AVR AT Mega 8* dengan kontroler *fuzzy*.

3.4 Pengujian dan Analisa Data

Setelah semua komponen pada alat sudah terhubung sesuai dengan diagram blok sistem yang telah dirancang dan perangkat lunak untuk mendukung sistem telah dibuat, maka diadakan pengujian dan analisa alat. Metode pengujian alat adalah sebagai berikut

- Menguji pada tiap-tiap blok rangkaian.
- Menguji sensor yang digunakan.
- Menggabungkan beberapa blok rangkaian menjadi sebuah sistem.
- Menguji sistem secara keseluruhan.
- Menganalisa hasil pengujian sistem secara keseluruhan.

3.5 Pengambilan Kesimpulan

Kesimpulan diambil berdasarkan data yang diperoleh dari pengujian sistem secara keseluruhan. Jika hasil yang didapatkan telah sesuai dengan yang direncanakan sebelumnya, maka sistem kendali tersebut telah berhasil memenuhi harapan dan tentunya memerlukan pengembangan lebih lanjut untuk penyempurnaan.



BAB IV

PERANCANGAN DAN PEMBUATAN ALAT

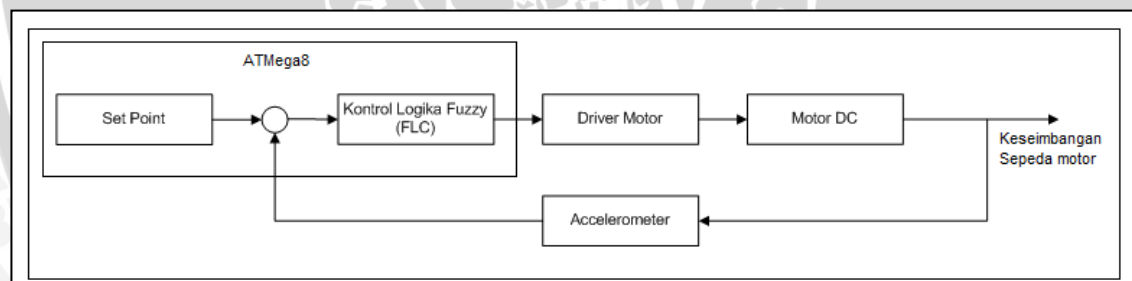
4.1 Perancangan Sistem

Perancangan alat ini dilakukan secara bertahap yaitu blok demi blok, sehingga akan memudahkan dalam menganalisa setiap bloknnya maupun secara keseluruhan. Perancangan dan pembuatan sistem ini terdiri atas dua bagian yaitu perancangan perangkat keras (*hardware*) dan perancangan perangkat lunak (*software*). Perancangan ini meliputi :

- Cara kerja alat.
- Spesifikasi alat.
- Perancangan sistem mekanik AKI.
- Perancangan sistem elektrik AKI.
- Perancangan algoritma kontroler *fuzzy* pada AT Mega 8.

4.1.1 Blok Diagram

Untuk mempermudah pengerjaan alat maka perlu adanya diagram blok umum sistem yang ditunjukkan dalam Gambar 4.1 :



Gambar 4.1 Blok Diagram Sistem AKI

Keterangan dari blok diagram di atas adalah sebagai berikut :

- *Accelerometer* berfungsi untuk mengukur kemiringan miniatur sepeda motor terhadap titik tegaknya. *Accelerometer* yang digunakan adalah MMA7455 3-Axis Accel Module.
- Pusat pengendalian sistem menggunakan Mikrokontroler ATMega8 yang memberikan keluaran berupa *duty cycle* PWM kepada *driver* motor.

- Motor yang digunakan adalah motor DC, yang berfungsi sebagai *pemutardisc inertia*.
- *Driver* motor menggunakan *driver* PWM mosfet IRFZ44 dan relay *Dual Pole Dual Throw* (DPDT).

4.1.2 Prinsip Kerja Alat

Miniatur sepeda motor dikatakan seimbang apabila *accelerometer* memberikan tegangan yang menunjukkan dimana miniatur sepeda motor tegak lurus terhadap permukaan bumi. Apabila miniatur sepeda motor tidak tegak lurus terhadap garis normal atau dalam posisi miring maka kontroler akan memberi perintah pada *driver* motor untuk mengendalikan motor untuk memutar *disc inertia* sesuai arah kemiringan miniatur sepeda motor sehingga keadaan miniatur sepeda motor tetap seimbang.

Masukan kontroler berupa kemiringan, sementara keluaran kontroler adalah berupa *duty cycle* PWM yang kemudian diteruskan ke-*driver* motor dan menghasilkan tegangan untuk memutar motor DC dengan kecepatan dan arah tertentu.

4.2 Perancangan Perangkat Keras (Hardware)

Berdasarkan diagram blok perancangan alat yang telah disusun, perancangan perangkat keras meliputi perancangan mekanik, *accelermeter* dan *driver* motor serta penggunaan modul *AT Mega 8*. Di bawah ini adalah penjelasan masing-masing rangkaian penyusun keseluruhan alat.

4.2.1 Accelerometer

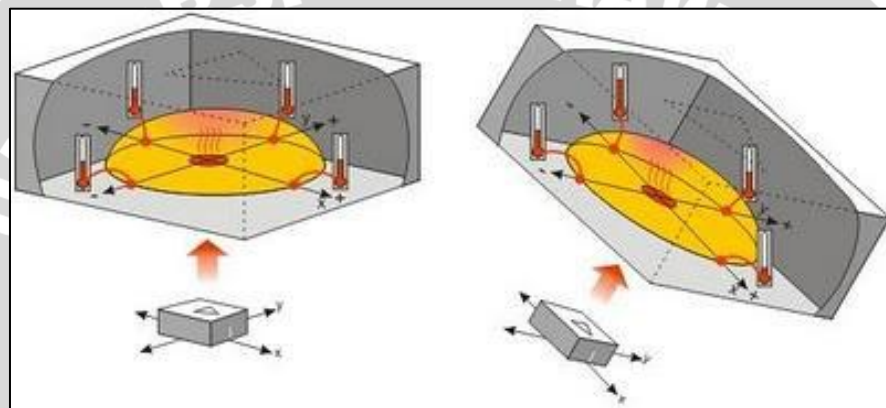
Accelerometer digunakan sebagai pendeteksi kemiringan miniatur sepeda motor terhadap permukaan bumi. Setiap perubahan sudut pada miniatur sepeda motor akan terdeteksi oleh *accelerometer* yang kemudian akan mengeluarkan tegangan sesuai dengan besaran kemiringan sudut miniatur sepeda motor terhadap permukaan bumi.

Gambar 4.2 adalah *accelerometer* MMA7455 3-Axis Accel Module yang digunakan pada miniatur sepeda motor.



Gambar 4.2 Accelerometer MMA7455 3-Axis Accel Module

Prinsip kerja sensor Accelerometer seperti yang ditunjukkan pada Gambar 4.3.



Gambar 4.3 Prinsip Kerja Accelerometer

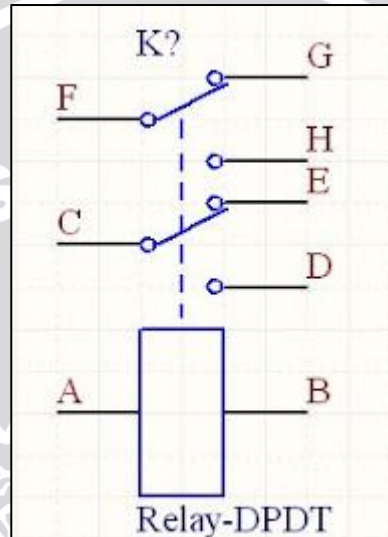
Sensor menghasilkan panas yang mana jika dimiringkan maka panas tersebut akan memanasi bagian pinggir sensor yang merupakan bahan *thermocoupler*. Sehingga jika dipanaskan akan menghasilkan perubahan dan perbedaan panas yang mengakibatkan terjadinya perubahan tegangan yang nantinya akan dimasukkan kedalam ADC pada mikrokontroler.

4.2.2 Perancangan Driver Motor

Untuk *driver* motor DC digunakan PWM Mosfet IRFZ44 dan *relay dual pole dual throw (DPDT)*. Mosfet IRFZ44 digunakan sebagai penguat PWM yang masuk ke *driver* untuk diteruskan ke motor DC dan *relay dual pole dual throw* digunakan untuk mengatur arah putaran motor.

Relay terdiri dari *Coil & Contact*. *Coil* adalah gulungan kawat yang mendapat arus listrik, sedang *contact* adalah sejenis saklar yang pergerakannya tergantung dari ada tidaknya arus listrik di *coil*.

Contact ada 2 jenis : *Normally Open* (kondisi awal sebelum diaktifkan *open*), dan *Normally Closed* (kondisi awal sebelum diaktifkan *close*). Secara sederhana berikut ini prinsip kerja dari relay, ketika *Coil* mendapat energi listrik (*energized*), akan timbul gaya elektromagnet yang akan menarik *armature* yang berpegas, dan *contact* akan menutup. Skema *relay DPDT* dapat dilihat pada Gambar 4.4.

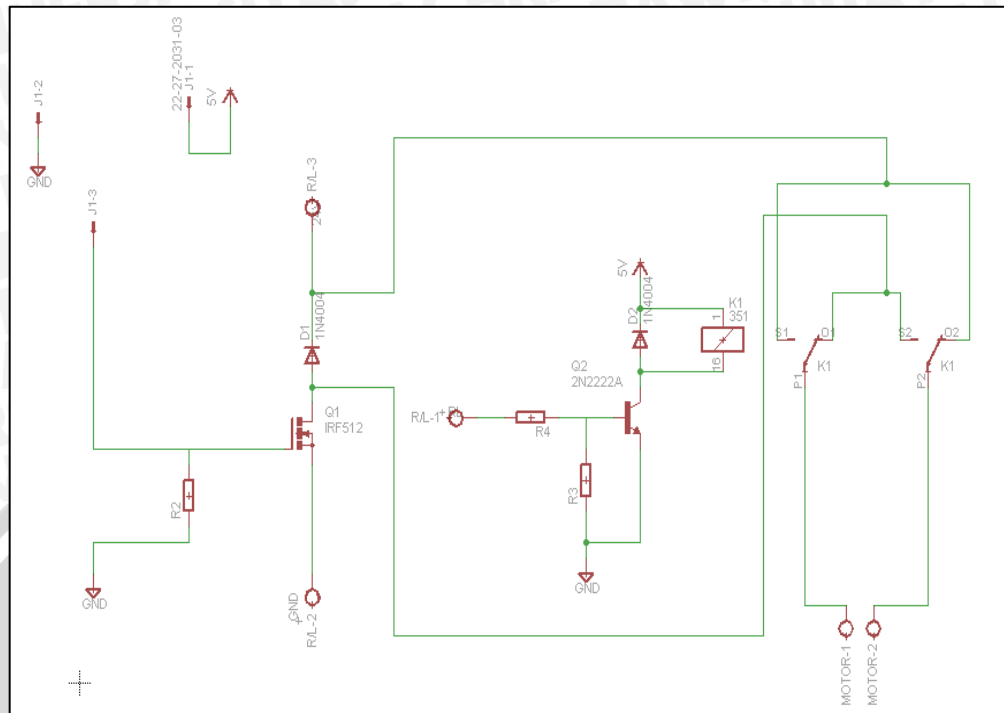


Gambar 4.4 Skema Prinsip Dasar Kerja Relay DPDT

Blok AB terdapat kumparan sebagai *driver*. Prinsip kerja dari relay ini yaitu, ketika AB belum dilewati arus, maka terminal CE dan FG akan tersambung, dan pada saat AB dilewati arus maka plat C dan F akan berpindah sehingga terminal CD dan FH akan tersambung. Jadi ketika *driver* AB dilewati arus plat yang berpindah ada 2 yaitu C dan F.

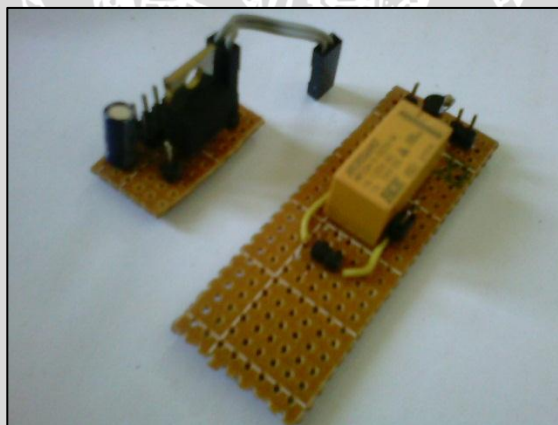
Alasan pemilihan IRFZ44N sebagai komponen *driver* yang akan digunakan adalah karena IRFZ44N memiliki $I_{D\ max}$ cukup besar dan $R_{DS\ on\ max}$ lebih kecil dibanding beberapa jenis E-MOSFET kanal N lain, contohnya IRF520, IRF540, IRF630, IRF720, IRF820 dan IRFZ40. Dengan $I_{D\ max}$ yang cukup besar dan $R_{DS\ on\ max}$ yang kecil membuat IRFZ44N tidak cepat panas ketika dilewati arus yang cukup besar.

Rangkaian elektrik *driver motor* relay dan mosfet yang digunakan untuk mengatur motor DC dapat dilihat pada Gambar 4.5



Gambar 4.5 Rangkaian Driver Motor DC.

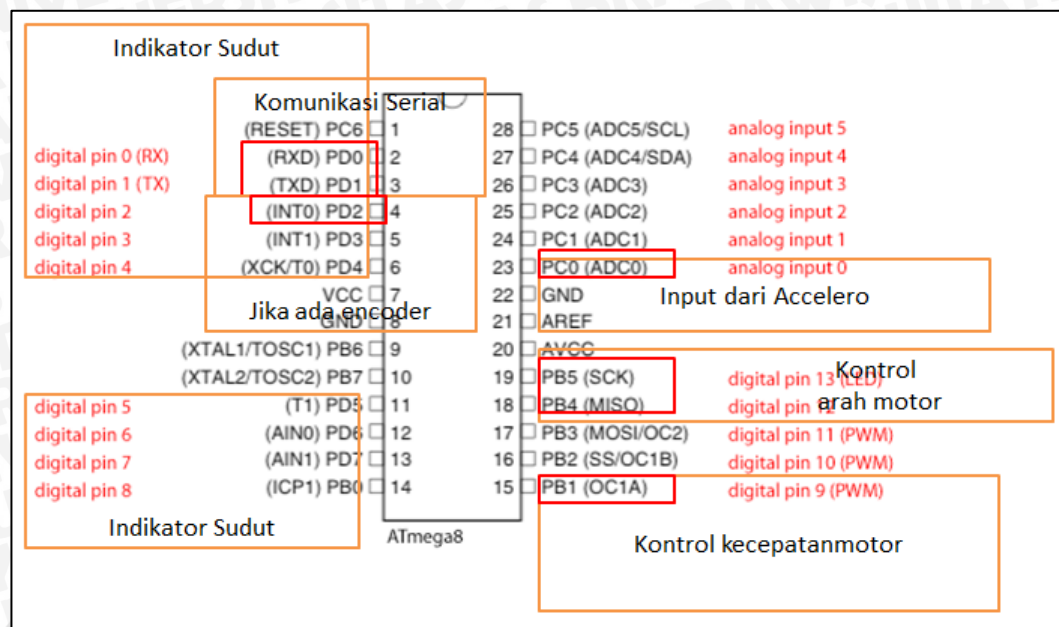
Gambar 4.6 adalah gambar implementasi *driver* motor DC pada miniatur sepeda motor.



Gambar 4.6 Driver Motor DC

4.2.3 Rangkaian Mikrokontroler AT Mega 8

Pada miniatur sepeda motor ini digunakan Mikrokontroler AT Mega 8 sebagai pusat dari pengolahan data. Skematik Mikrokontroler AT Mega 8 dan perancangan secara keseluruhan ditunjukkan pada Gambar 4.7.



Gambar 4.7 Mikrokontroler ATmega8

Agar sebuah Mikrokontroler ATmega8 dapat bekerja sebagai pengontrol, kaki-kaki/pin Mikrokontroler ATmega 8 dihubungkan dalam rangkaian-rangkaian eksternal. Dalam perancangan ini, port yang digunakan adalah sebagai berikut :

1. Port PB5 (SPI Bus Master clock Input) dan PB (SPI Bus Master Input/Slave Output)

Port ini terhubung dengan motor DC pemutar disc inertia sebagai pengatur arah putar motor.

2. Port PB1 (Timer/Counter1 Output Compare Match A Output)

Port ini difungsikan sebagai keluaran pulse width modulation (PWM) berfungsi sebagai kontrol kecepatan motor yang memutar disc inertia agar miniatur sepeda motor tetap seimbang.

3. Port PC0 (ADC Input Chanel 0)

Port ini sebagai masukkan dari accelerometer. Port ini dapat digunakan untuk mengubah input yang berupa tegangan analog menjadi data digital.

4. Port PD0 (USART Input Pin) dan PD1 (USART Output Pin)

Merupakan jalur data komunikasi serial dengan level sinyal TTL. Pin TXD (USART Output Pin) berfungsi untuk mengirimkan data

serial, sedangkan RXD (*USART Input Pin*) kebalikannya yaitu sebagai pin yang berfungsi untuk menerima data serial.

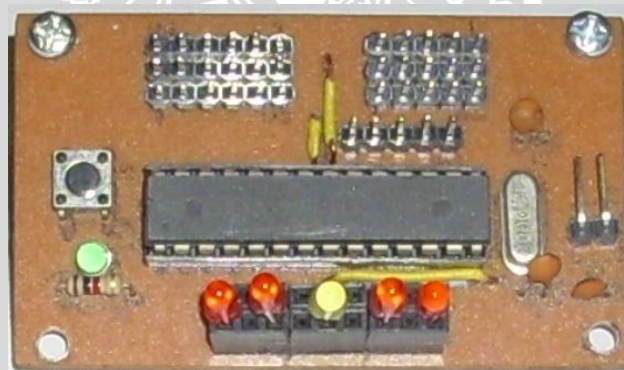
5. *Port PD5 (Timer/Counter 1 External Clock Input), PD6 (Analog Comparator Positive Input), PD7 (Analog Comparator Negative Input), PB0 (Timer/Counter1 Input Capture Pin).*

Berfungsi sebagai indikator sudut ditunjukkan oleh lampu *Light Emmiting Diode* (LED) yang di pasang pada perangkat elektronik miniatur sepeda motor

6. *Port VCC (Digital supply voltage) dan GND (Ground)*

Port VCC merupakan port suplai tegangan digital. Besarnya tegangan berkisar antara 4,5 – 5,5V. Dan GND adalah Referensi nol suplai tegangan digital.

Gambar 4.8 merupakan gambar modul mikrokontroler ATmega8 yang digunakan pada miniatur sepeda motor.



Gambar 4.8 Modul Mikrokontroler ATmega8

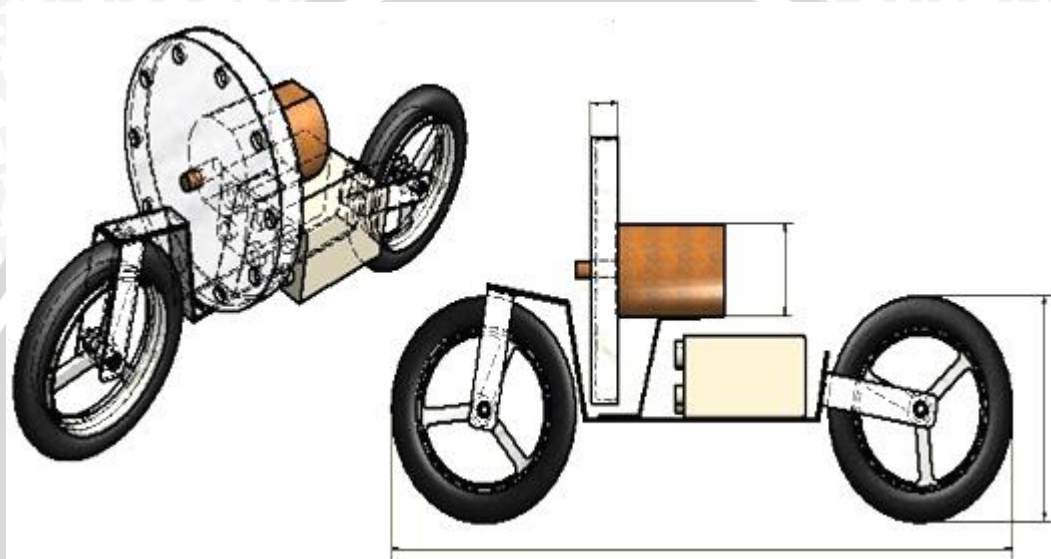
Selanjutnya, modul Mikrokontroler ATmega8 dihubungkan dengan komputer dengan menggunakan komunikasi *parallel*. Komunikasi ini bertujuan agar dapat dilakukan pemrograman di komputer dan selanjutnya program tersebut dapat ditanamkan pada Mikrokontroler ATmega8.

4.2.4 Perancangan Mekanik Miniatur Sepeda Motor

Miniatur sepeda motor mempunyai beberapa bagian yang penting dalam perancangannya, yaitu: bentuk dan posisi *disc inertia*, posisi motor DC dan *accelerometers* serta kerangka miniatur sepeda motor. Motor DC dipasang ditengah

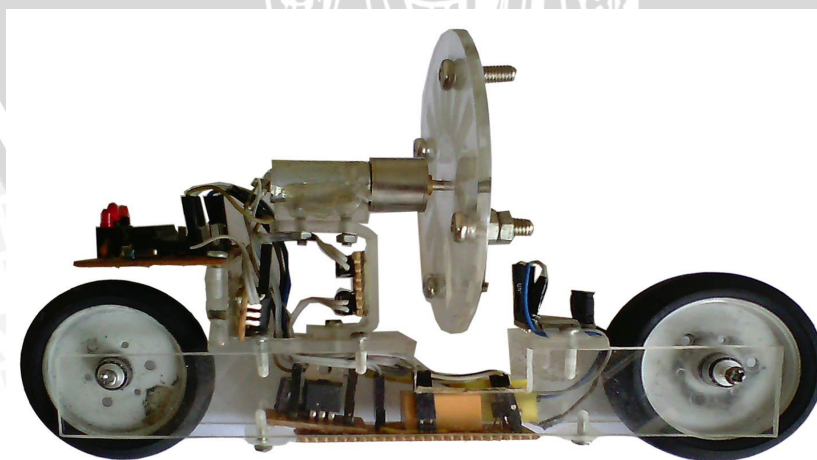
dan sejajar dengan miniatur sepeda motor agar posisi *disc inertia* berpotongan dengan posisi motor. Sedangkan posisi dari *accelerometer* yaitu dipasangkan sejajar dengan posisi pada saat motor berada dalam keadaan seimbang, sehingga setiap perubahan posisi miniatur sepeda motor akan terbaca oleh *accelerometer*.

Rencana desain awal mekanik yang akan dibuat pada penelitian ini ditunjukkan pada Gambar 4.9.



Gambar 4.9 Implementasi Desain Bagian Pertama “Posisi *Disc Inertia* dan Posisi Sensor *Accelerometer*”

Gambar 4.10 berikut adalah miniatur sepeda motor beserta komponen elektrik, motor DC dan mekanis penyeimbang *disc inertia*.



Gambar 4.10 Miniatur Sepeda Motor dan Mekanis Penyeimbang

4.2.4.1 Spesifikasi Alat

Spesifikasi alat yang dibuat adalah sebagai berikut:

- Sistem penyeimbang menggunakan *disc inertia* dengan diameter 8.5mm & tebal 3mm.
- Menggunakan catu daya 5V untuk catu *accelerometer* & mikrokontroler ATmega8.
- Menggunakan catu daya 12V untuk motor DC.
- Sensor kemiringan menggunakan *accelerometer*.
- Menggunakan ATmega 8 sebagai pengendali utama.
- *Driver* motor berupa PWM Mosfet IRFZ44 dan *relay DPDT*
- Diameter kedua roda miniatur sepeda motor 55mm.
- Panjang kerangka sepeda motor 160mm.

4.2.4.2 Pemilihan Motor DC

Motor yang dipakai pada alat harus memenuhi spesifikasi dari perancangan meliputi torsi dan kecepatannya. Torsi motor berhubungan dengan beban yang ditanggung oleh motor meliputi: berat *disc inertia* dan kekuatan motor untuk berputar ke arah yang berlawanan. Sedangkan kecepatan motor berhubungan dengan kecepatan jatuh dari posisi tegak lurus ke posisi miring atau tidak tegak.

Motor yang digunakan pada penelitian ini adalah motor DC 12 VDC 200 rpm. Seperti yang terlihat pada gambar 4.11.



Gambar 4.11 Motor DC gearbox 12 VDC 200rpm

Keuntungan utama motor DC adalah sebagai pengendali kecepatan, yang tidak mempengaruhi kualitas pasokan daya. Motor ini dapat dikendalikan dengan mengatur:

- Tegangan motor: Meningkatkan tegangan motor akan meningkatkan kecepatan

- Arus medan: Menurunkan arus medan akan meningkatkan kecepatan.

Hubungan antara kecepatan, flux medan dan tegangan dinamo ditunjukkan dalam Persamaan 4.1 dan 4.2.

$$\text{Gaya elektromagnetik: } E = K \cdot \Phi \cdot N \dots \dots \dots (4.1)$$

$$\text{Torque: } T = K \cdot \Phi \cdot I_a \dots \dots \dots (4.2)$$

Dimana:

E = gaya elektromagnetik yang dikembangkan pada terminal dinamo (volt)

Φ = flux medan yang berbanding lurus dengan arus medan

N = kecepatan dalam RPM (putaran per menit)

T = *torque* elektromagnetik

I_a = arus motor

K = konstanta persamaan

Secara umum, kecepatan rotasi dari motor DC adalah sebanding dengan tegangan yang diterapkan untuk itu, dan torsi adalah sebanding dengan arus. Kontrol kecepatan dapat dicapai dengan baterai variabel, tegangan suplai variabel, resistor atau kontrol elektronik. Arah dari putaran motor DC dapat berubah dengan membalik kutub tegangan.

4.3 Perancangan Perangkat Lunak (software)

4.3.1 Perancangan Kontroler Logika Fuzzy

4.3.1.1 Menentukan Variabel Masukan dan Variabel Keluaran

Variabel masukan untuk kontrol *fuzzy* terdapat dua yaitu kemiringan (*error*) (e) dan perubahan kemiringan (*derror*) (de/dt) sebagai masukan pembanding dan variable keluaran berupa % *duty cycle* untuk PWM motor DC.

4.3.1.2 Fuzzifikasi

Fuzzifikasi adalah proses mengubah nilai *input crisp* menjadi *input fuzzy* melalui fungsi keanggotaan. Untuk menentukan derajat keanggotaan suatu *input craps*, tarik garis vertikal yang melalui *input crisp* tersebut (sumbu x). Kemudian tentukan nilai koordinat sumbu y yang terpotong oleh garis vertikal tersebut. Nilai koordinat sumbu y tersebut menyatakan derajat keanggotaan *input crisp* pada suatu label.

4.3.1.3 Membership Function Input

4.3.1.3.1 Membership Function Kemiringan (error (e))

Fungsi keanggotaan kemiringan terdiri dari (B, NZ, Z, PZ, P) label,. Fungsi keanggotaan label N dan P berbentuk trapesium, sedangkan untuk label NZ, Z, PZ berbentuk segitiga. Hal ini dikarenakan miniatur sepeda motor dijaga supaya tetap berada pada kondisi tegak, jadi ketika ada sedikit saja terjadi kemiringan atau perubahan pada variabel Z diharapkan aksi pengendalian segera bereaksi.

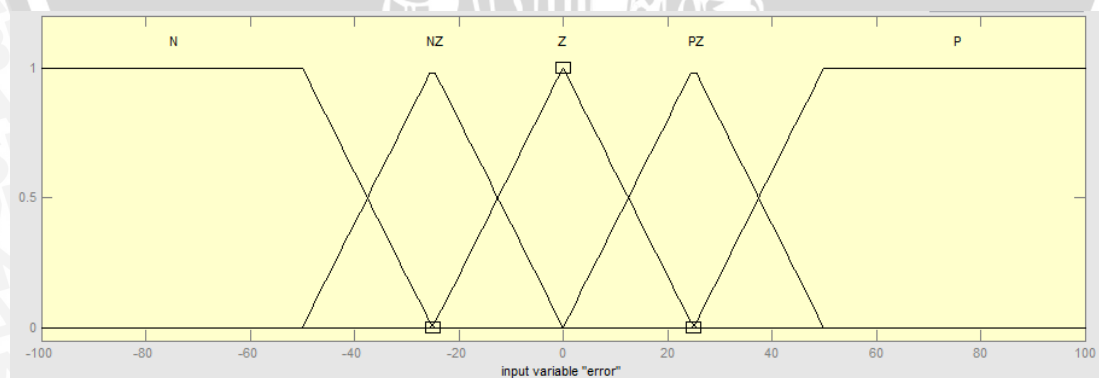
4.3.1.3.2 Membership Function Perubahan Kemiringan (derror (de/dt))

Fungsi keanggotaan perubahan kemiringan terdiri dari 5 label, yaitu N, NZ, Z, PZ, P. Fungsi keanggotaan label N dan P berbentuk trapesium, sedangkan untuk label NZ, Z, PZ berbentuk segitiga.

4.3.1.3.3 Menentukan batas Membership Function (MF) pada Kemiringan dan Perubahan Kemiringan

Berdasarkan keinginan perancang yang mengharapkan respon kontroler cepat, maka penentuan membership *fuzzy* dilakukan dengan memaksimalkan kinerja kontroler pada kondisi miniatur sepeda motor pada saat mulai tidak tegak (kemiringan kecil), karena pada saat kemiringan kecil miniatur sepeda motor akan mudah dikendalikan, sebaliknya saat kemiringannya besar miniatur sepeda motor akan sulit dikendalikan lagi.

Fungsi keanggotaan kemiringan ditunjukkan pada gambar 4.12.

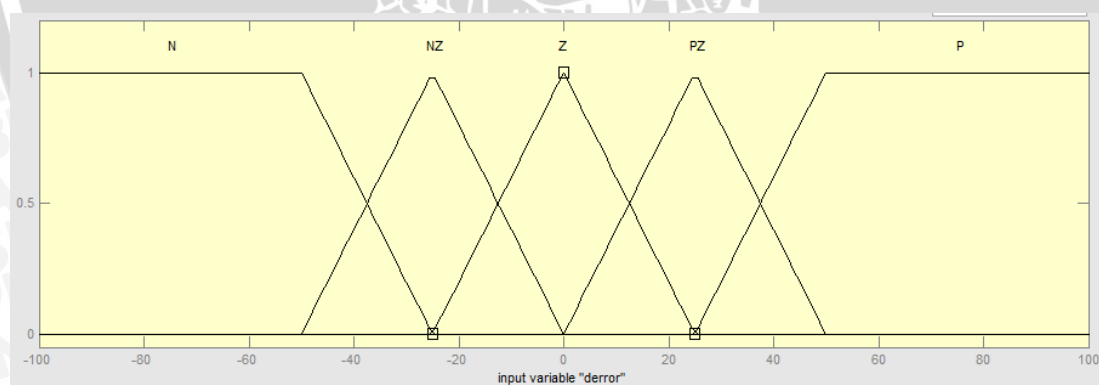


Gambar 4.12 Fungsi Keanggotaan Kemiringan

Dengan sumbu x merupakan besarnya kemiringan dan sumbu y merupakan nilai derajat keanggotaan kemiringan sementara label kemiringan dijelaskan sebagai berikut :

- N : *Negatif* (Variabel kemiringan kearah kiri (besar) dengan bentuk fungsi keanggotaan setengah trapesium dengan nilai keanggotaan - 50, -25)
- NZ : *Negatif Zero* (Variabel kemiringan kearah kirih (kecil) dengan bentuk fungsi keanggotaan segitiga dengan nilai keanggotaan -50, -25, -0)
- Z : *Zero* (Variabel posisi tengah dengan bentuk fungsi keanggotaan segitiga dengan nilai keanggotaan -25, 0, 25)
- PZ : *Positif Zero* (Variabel kemiringan kearah kanan (kecil) dengan bentuk fungsi keanggotaan segitiga dengan nilai keanggotaan 0, 25, 50)
- P : *Positif* (Variabel kemiringan kearah kanan (besar) dengan bentuk fungsi keanggotaan setengah trapesium dengan nilai keanggotaan 25, 50).

Fungsi keanggotaan perubahan kemiringan ditunjukkan pada gambar 4.13.

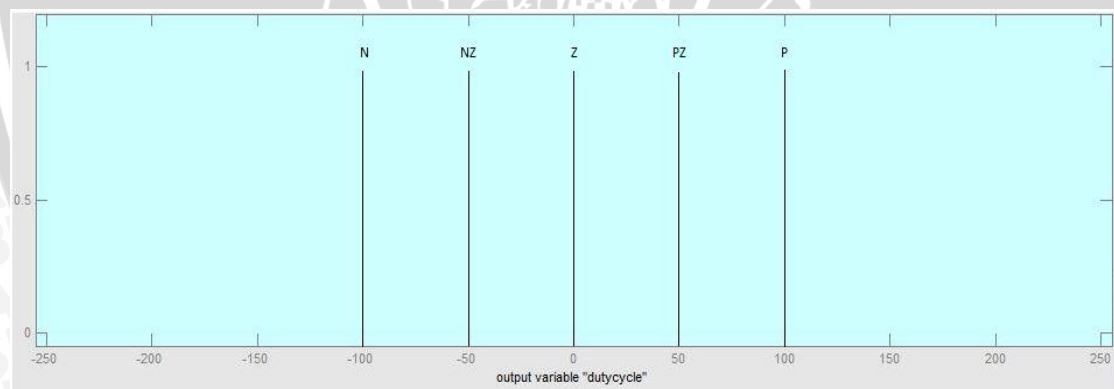


Gambar 4.13 Fungsi Keanggotaan Perubahan Kemiringan

Dengan sumbu x merupakan besarnya perubahan kemiringan dan sumbu y merupakan nilai derajat keanggotaan kemiringan sementara label kemiringan dijelaskan sebagai berikut ::

- N : *Negatif* (Variabel kemiringan kearah kiri (besar) dengan bentuk fungsi keanggotaan setengah trapesium dengan nilai keanggotaan -50, -25)
- NZ : *Negatif Zero* (Variabel kemiringan kearah kirih (kecil) dengan bentuk fungsi keanggotaan segitiga dengan nilai keanggotaan -50, -25, -0)
- Z : *Zero* (Variabel posisi tengah dengan bentuk fungsi keanggotaan segitiga dengan nilai keanggotaan -25, 0, 25)
- PZ : *Positif Zero* (Variabel kemiringan kearah kanan (kecil) dengan bentuk fungsi keanggotaan segitiga dengan nilai keanggotaan 0, 25, 50)
- P : *Positif* (Variabel kemiringan kearah kanan (besar) dengan bentuk fungsi keanggotaan setengah trapesium dengan nilai keanggotaan 25, 50).

4.3.1.3.4 Menentukan Batas Membership Function (MF) Keluaran Pada Motor DC



Gambar 4.14 Fungsi Keanggotaan Keluaran Motor DC

Gambar 4.14 adalah gambar fungsi keanggotaan keluaran motor DC. Dengan sumbu x merupakan besarnya pulse / second untuk putaran motor DC dan sumbu y merupakan nilai derajat keanggotaan keluaran, sementara label keluaran dijelaskan sebagai berikut :

- N : *Negatif* (motor berputar berlawanan arah jarum jam dengan kecepatan *pulse* sebesar 100 *pulse / second*)

- NZ : *Negatif Zero* (motor berputar berlawanan arah jarum jam dengan kecepatan *pulse* sebesar 50 *pulse / second*)
- Z : *Zero* (motor tidak berputar)
- PZ : *Positif Zero* (motor berputar searah jarum jam dengan kecepatan *pulse* sebesar 50 *pulse / second*)
- P : *Positif* (motor berputar searah jarum jam dengan kecepatan *pulse* sebesar 100 *pulse / second*)

Pada skripsi ini alasan pemilihan bentuk *membership function* keluaran menggunakan 5 label agar kontroler dapat dengan cepat memberikan respon terhadap motor DC untuk menegakkan miniatur sepeda motor.

4.3.1.4 Kaidah Atur Kontrol Logika Fuzzy

Kaidah atur kontroler logika fuzzy didasarkan pada pengamatan dalam bentuk Jika-Maka. Setelah *crisp input* diubah menjadi *input fuzzy*, selanjutnya diolah sesuai dengan kaidah aturnya. Metode penalaran fuzzy yang dipergunakan oleh kontroler logika fuzzy adalah metode MAX-MIN. Kaidah atur selengkapnya terdapat dalam table 4.1 berikut :

Tabel 4.1 Tabel Kaidah Atur Kemiringan

ERROR dERROR	N	NZ	Z	PZ	P
N	N	N	N	NZ	PZ
NZ	N	N	NZ	Z	PZ
Z	N	NZ	Z	PZ	P
PZ	NZ	Z	PZ	P	P
P	NZ	PZ	P	P	P

Keterangan Tabel 4.1 :

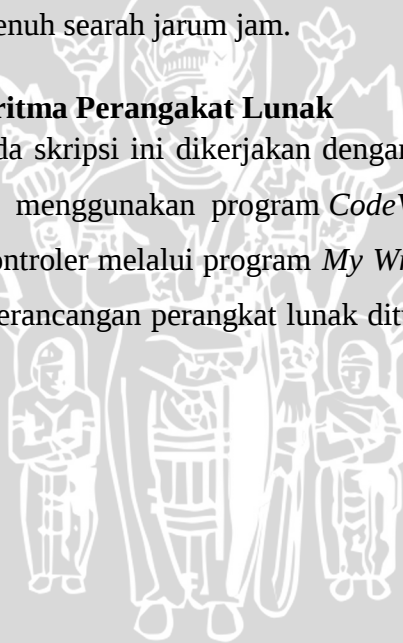
- Jika ERROR adalah N dan dERROR adalah N, maka motor akan berputar dengan kecepatan penuh berlawanan arah jarum jam.
- Jika ERROR adalah N dan dERROR adalah NZ, maka motor akan berputar dengan kecepatan penuh berlawanan arah jarum jam.
- Jika ERROR adalah N dan dERROR adalah Z, maka motor akan berputar dengan kecepatan penuh berlawanan arah jarum jam.

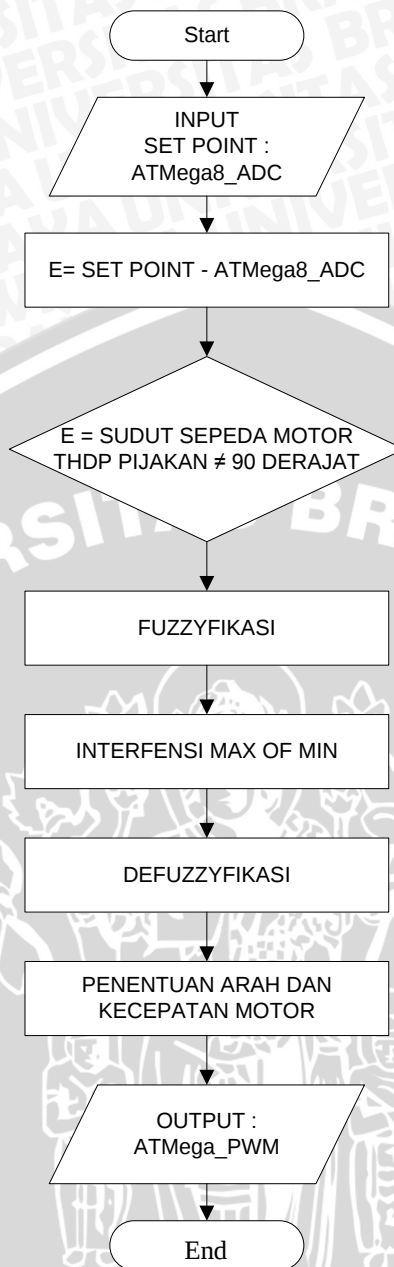
.....
”dan seterusnya”.....

- Jika ERROR adalah P dan dERROR adalah P, maka motor akan berputar dengan kecepatan penuh searah jarum jam.

4.3.2 Perancangan Algoritma Perangkat Lunak

Perangkat lunak pada skripsi ini dikerjakan dengan bahasa pemrograman C++ dan di *compile* dengan menggunakan program *CodeVisionAVR Pro v2.03.9*. Serta dituliskan ke mikrokontroler melalui program *My Write*. Kontroler Kontrol Logika Fuzzy. *Flowchart* perancangan perangkat lunak ditunjukkan pada Gambar 4.15.





Gambar 4.15 Flowchart program

BAB V

PENGUJIAN DAN ANALISIS SISTEM

Tujuan pengujian sistem ini adalah untuk menentukan apakah alat yang telah dibuat berfungsi dengan baik dan sesuai dengan perancangan. Pengujian pada sistem ini meliputi pengujian setiap blok maupun pengujian secara keseluruhan. Pengujian setiap blok ini dilakukan untuk menemukan letak kesalahan dan mempermudah analisis pada sistem apabila alat tidak bekerja sesuai dengan perancangan. Pengujian pada sistem ini dibagi menjadi empat bagian, yaitu:

- Pengujian motor DC
- Pengujian *driver* Motor DC
- Pengujian *Accelerometer*
- Pengujian keseluruhan sistem

5.1 Pengujian Motor DC

5.1.1 Tujuan

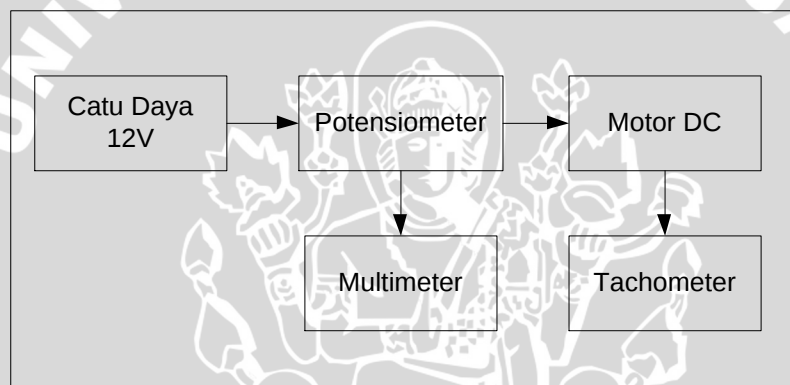
Mengetahui *output* dari driver motor apabila diberi *input* yang berbeda-beda, serta mengetahui waktu yang dibutuhkan motor untuk mencapai kecepatan maksimum.

5.1.2 Peralatan yang digunakan

- Mikrokontroler ATmega 8
- Motor DCgearbox
- Komputer
- *Software CodeVisionAVR Pro v2.03.9.*
- Catu daya 12 Volt
- Potensiometer
- Serial port max 232
- *Tachometer*

5.1.3 Langkah pengujian

1. Merangkai alat seperti pada Gambar 5.1.
2. Membuat program untuk menentukan arah dan kecepatan motor pada *software CodeVisionAVR Pro v2.03.9*.
3. *Load Flash* program yang telah dibuat, hapus program yang ada dalam mikrokontroler ATmega8, tulis program yang telah dibuat ke dalam mikrokontroler ATmega8 menggunakan *software Code Vision Pro v.2.03.9* melalui kabel serial port ke PC.
4. Mengukur kecepatan pada motor yang terbaca oleh encoder melalui serial port yang terhubung ke komputer.
5. Menggambar data yang didapatkan pada grafik.



Gambar 5.1 Diagram Blok Pengujian Motor

Gambar 5.1 menunjukkan diagram blok pengujian motor DC. Catu daya yang digunakan adalah catu daya sebesar 12 volt. Catu daya ini dipilih berdasarkan spesifikasi motor yang dapat bekerja dengan tegangan maksimal sebesar 12 volt. Tegangan dari catu daya diatur menggunakan sebuah potensiometer mulai dari 0 volt hingga 12 volt dengan interval sebesar 0.5 volt. Motor akan berputar dengan kecepatan tertentu dan diukur dengan menggunakan sebuah *tachometer*. Dengan menggunakan tegangan maksimal 12 volt, waktu yang digunakan motor untuk mencapai kecepatan maksimum akan diukur menggunakan *tachometer*. Gambar 5.2 menunjukkan data dari pengujian motor. Tabel 5.1 menunjukkan data pengukuran waktu motor DC mencapai kecepatan maksimum dan Tabel 5.2 menunjukkan data pengukuran motor DC.

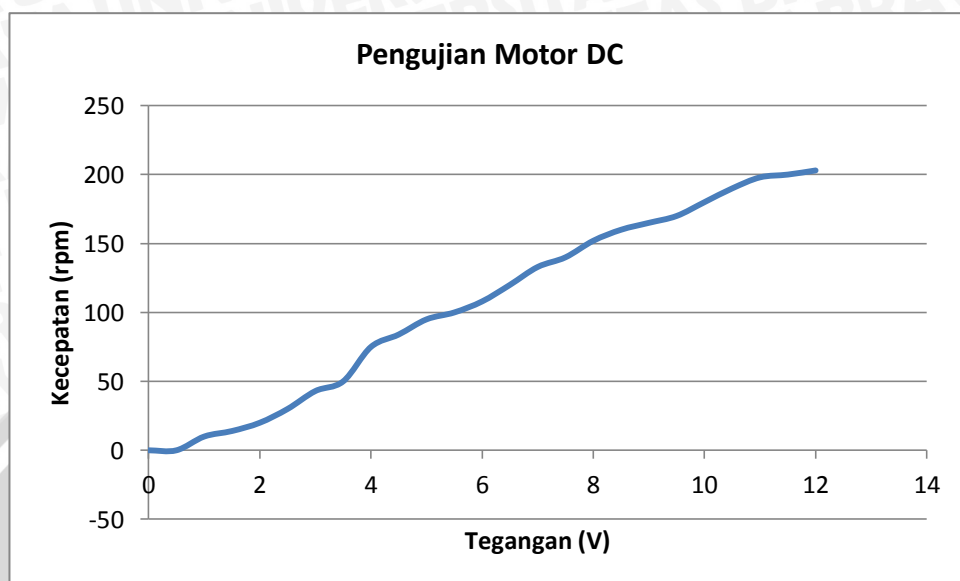
Tabel 5.1 Data Pengukuran Waktu Motor DC Mencapai Kecepatan Maksimum

No. Percobaan	Waktu (detik)	Kecepatan (rpm)
1	4.46	201
2	4.4	203
3	4.48	202
4	4.3	203
5	4.8	203
6	4	202
7	4.4	203
8	4.2	201
9	4.4	203
10	4.36	203
Rata - rata	4.376	202.5

Tabel 5.2 Data Pengukuran Kecepatan Motor DC

Tegangan Motor DC (V)	Kecepatan Motor DC (rpm)	Tegangan Motor DC (V)	Kecepatan Motor DC (rpm)
0	0	6.5	120
0.5	0	7	133
1	10	7.5	140
1.5	14	8	149
2	20	8.5	155
2.5	30	9	165
3	43	9.5	169
3.5	50	10	175
4	75	10.5	185
4.5	84	11	198
5	95	11.5	200
5.5	100	12	203
6	108		

Gambar 5.2 menunjukkan grafik kecepatan putar motor DC sesuai dengan pengukuran.



Gambar 5.2 Grafik Pengukuran Kecepatan Motor DC

Dari hasil pengujian dapat dilihat bahwa motor dapat berputar pada saat tegangan minimum 1V volt, motor DC *gearbox* ini dapat berputar pada range tegangan yang relatif kecil. Dan dari grafik terlihat hubungan antara tegangan dan kecepatan motor DC hampir linier. Sehingga dapat diambil kesimpulan bahwa motor DC *gearbox* ini sudah baik untuk digunakan dalam penelitian ini.

5.2 Pengujian Driver Motor

5.2.1 Tujuan

Untuk mengetahui *output* dari *driver* motor apabila diberikan input PWM yang berbeda.

5.2.2 Peralatan yang Digunakan

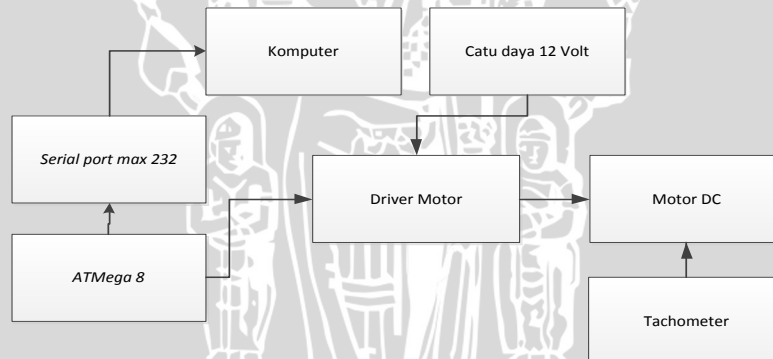
- Mikrokontroler ATmega 8
- *Driver* motor relay dan Mosfet.
- Komputer
- Serial port Max 232
- Motor DC
- Program CodeVisionAVR Pro v2.03.9
- Multimeter
- Catu daya 12 Volt

- Tachometer

5.2.3 Langkah Pengujian

1. Merangkai alat seperti pada Gambar 5.3.
2. Membuat program untuk menentukan arah dan kecepatan motor pada *software CodeVisionAVR Pro v2.03.9*
3. Mengukur tegangan pada kutub motor DC, untuk mengetahui tegangan keluaran dari *driver* motor yang masuk ke motor DC.
4. Menggambar data yang didapatkan pada grafik.

Gambar 5.3 menunjukkan diagram blok pengujian *driver* motor untuk melihat *output* serta *input* dari *driver* motor apakah sudah sesuai dengan yang diharapkan. *Driver* motor merupakan relay dan mosfet, yang dicatu oleh tegangan 12 V untuk memutar motor DC. Kemudian motor DC yang diputar diukur kecepatannya menggunakan *tachometer*. Sedangkan untuk mengatur besar kecilnya tegangan PWM digunakan serial port max 232 yang terhubung ke computer dan kemudian diatur melalui program *CodeVisionAVR Pro v2.03.9*.



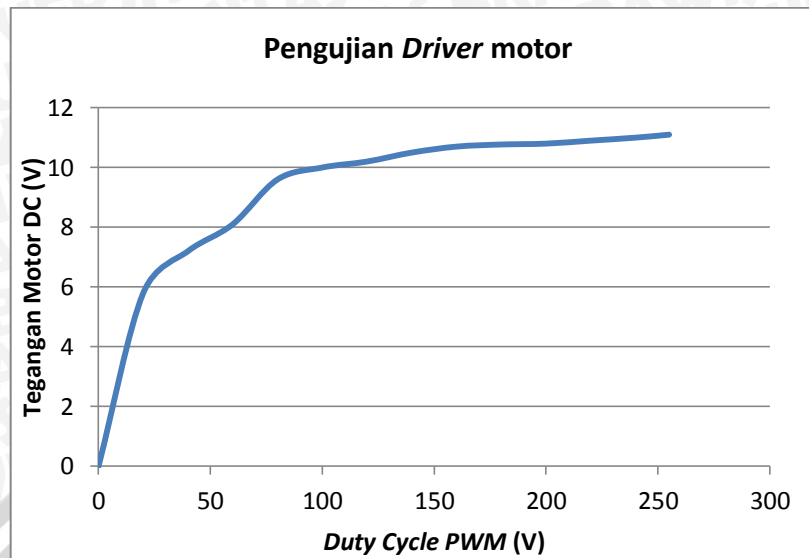
Gambar 5.3 Diagram Blok Pengujian *Driver* Motor

Pada Tabel 5.2 dapat dilihat data hasil pengukuran pada *driver* motor yang dilakukan dengan langkah-langkah pengujian. Dan selanjutnya pada Gambar 5.4 menunjukkan grafik hubungan antara *duty cycle* PWM yang masuk ke *driver* motor dengan tegangan yang dikeluarkan oleh *driver* driver motor. Serta Gambar 5.5 menunjukkan grafik hubungan antara *duty cycle* PWM dengan kecepatan motor DC.

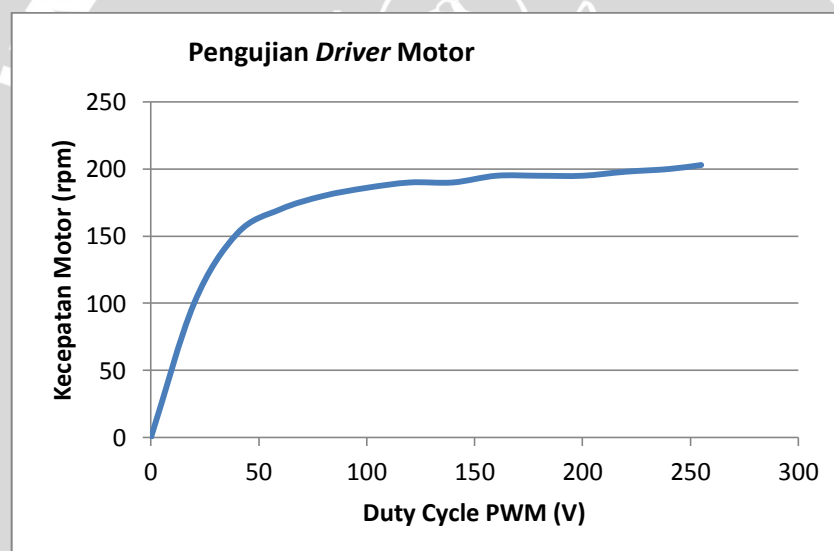
Dari Tabel 5.3 terlihat bahwa motor akan diam apabila diberi tegangan 0V sedangkan arah putaran motor bergantung pada Pin PB5. Pin PB 5 yang mengatur *on-off* dari *relay* pada *driver* motor, saat PB 5 berlogika 0 dan tegangan diberikan maka motor akan berputar ke arah kiri dan pada saat PB 5 berlogika 1 atau *relay* pada keadaan *on* maka motor akan berputar ke arah kanan.

Tabel 5.3 Data Pengukuran *Driver* Motor

PIN PB5	PWM (PIN PB1)	Tegangan Keluaran (V)	Kecepatan Motor (rpm)	Arah Putaran motor
0	0	0	0	Diam
0	20	5.8	100	Kiri
0	40	7.2	152	Kiri
0	60	8.1	170	Kiri
0	80	9.6	180	Kiri
0	100	10	186	Kiri
0	120	10.2	190	Kiri
0	140	10.5	190	Kiri
0	160	10.7	195	Kiri
0	180	10.77	195	Kiri
0	200	10.8	195	Kiri
0	220	10.9	198	Kiri
0	240	11	200	Kiri
0	255	11.1	203	Kiri
1	0	0	0	Diam
1	20	5.8	100	Kanan
1	40	7.2	152	Kanan
1	60	8.1	170	Kanan
1	80	9.6	180	Kanan
1	100	10	186	Kanan
1	120	10.2	190	Kanan
1	140	10.5	190	Kanan
1	160	10.7	195	Kanan
1	180	10.77	195	Kanan
1	200	10.8	195	Kanan
1	220	10.9	198	Kanan
1	240	11	200	Kanan
1	255	11.1	203	Kanan



Gambar 5.4 Grafik Hubungan *Duty Cycle PWM* dan Tegangan Motor DC



Gambar 5.5 Grafik Hubungan *Duty Cycle PWM* dan Kecepatan Motor DC

Dari Gambar 5.4 dan Gambar 5.5 terlihat bahwa hubungan antara *input* PWM dan *output* tegangan serta kecepatan motor DC relatif linier. Sehingga dapat dikatakan bahwa *driver* motor bekerja dengan cukup baik dan dapat digunakan pada pengontrolan motor DC pada miniatur sepeda motor.

5.3 Pengujian Accelerometer

5.3.1 Tujuan

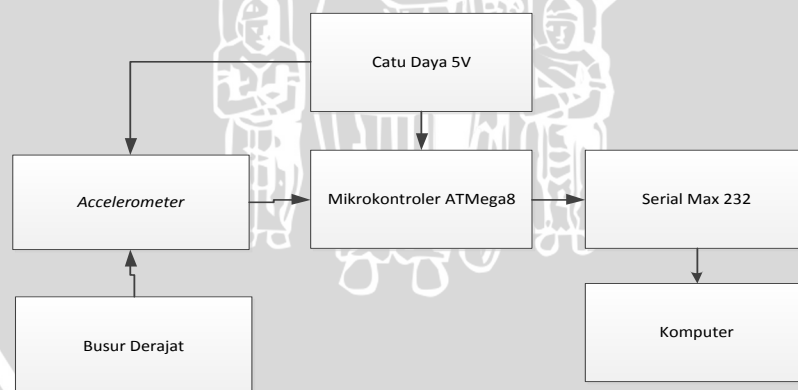
Mengetahui tingkat keakuratan dari *Accelerometer* dalam membaca perubahan sudut dari miniatur sepeda motor.

5.3.2 Peralatan yang Digunakan

- Sistem mekanik miniatur sepeda motor yang terhubung dengan *accelerometer* dan mikrokontroler ATmega8
- Penggaris busur.
- Catu daya 5 volt.
- Komputer.

5.3.3 Langkah pengujian

1. Memasang alat seperti pada Gambar 5.6.
2. Mengubah posisi kemiringan dari miniatur sepeda motor dan mengukur sudut dengan menggunakan penggaris busur
3. Mengukur tegangan keluaran *accelerometer* melalui program Visual Basic 6.
4. Menggambar data tegangan keluaran *accelerometer* pada grafik untuk melihat kelinieran dari *accelerometer*.



Gambar 5.6 Diagram Blok Pengujian Accelerometer

Gambar 5.6 menunjukkan diagram blok pengujian *accelerometer*. *Accelerometer* terhubung dengan mekanik miniatur sepeda motor dan mikrokontroler ATmega8. Kemiringan dari miniatur sepeda motor diubah-

ubah lalu komputer akan menunjukkan perubahan tegangan dari *accelerometer*.

Tabel 5.4 menunjukkan pengamatan pada *accelerometer*.

Tabel 5.4 Menunjukkan Data Pengamatan *Accelerometer*

No.	Data				
	Tegangan accelerometer (Volt)	ADC	Sudut Accelerometer(⁰)	Sudut Aktual (⁰)	Error
1	1.275492	261	41	41	0
2	1.298876	266	44	43	1
3	1.315445	269	45	45	0
4	1.332014	273	47	47	0
5	1.348583	276	49	49	0
6	1.413242	289	56	55	1
7	1.655906	339	83	82	1
8	1.676876	343	85	85	0
9	1.697846	348	88	88	0
10	1.718816	352	90	90	0
11	1.769997	362	96	96	0
12	1.810996	371	100	100	0
13	1.851995	379	105	105	0
14	1.892994	388	109	108	1
15	1.933993	396	114	115	1
16	1.974992	404	118	118	0
17	2.015991	413	123	124	1
18	2.05699	421	127	128	1
19	2.097989	430	132	132	0
20	2.138988	438	136	137	1

Rata – rata kesalahan *Accelerometer* dapat dihitung dengan menggunakan rumus 5.1.

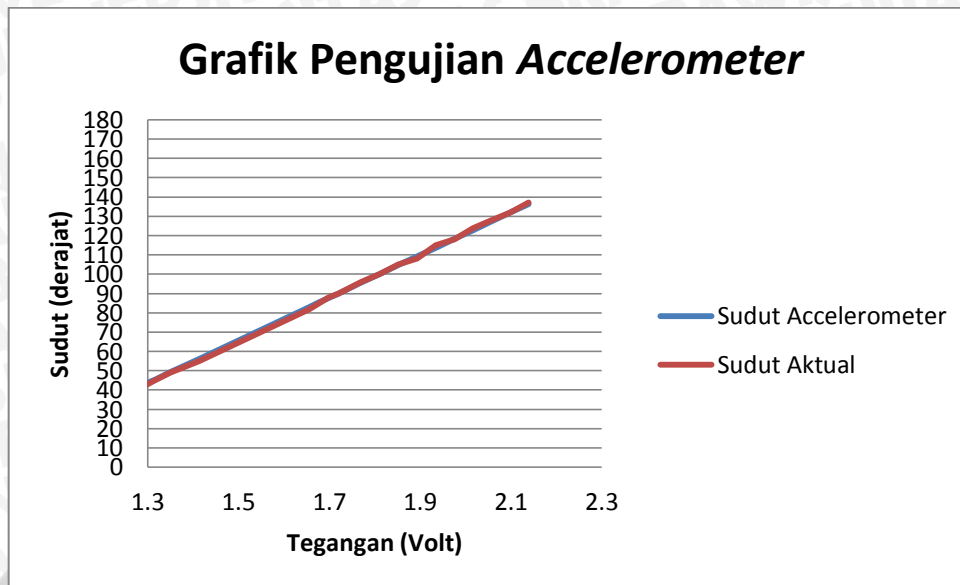
$$\text{rata – rata kesalahan (\%)} = \frac{\sum \text{error}}{\text{jumlah data}} \dots\dots\dots (5.1)$$

Sehingga didapat:

$$\text{rata- rata kesalahan (\%)} = \frac{8}{20} \dots\dots\dots (5.2)$$

$$\text{rata- rata kesalahan (\%)} = 0,4 \dots\dots\dots (5.3)$$

Gambar 5.7 menunjukkan grafik sudut pembacaan *Accelerometer* dan sudut aktual.



Gambar 5.7 Grafik Pengujian Accelerometer

Dari Tabel 5.4 dan Gambar 5.7 terlihat bahwa pembacaan sudut *Accelerometer* sudah akurat, terlihat dari selisih error dengan sudut aktual yang sangat kecil. Dapat disimpulkan bahwa *Accelerometer* sudah bekerja dengan baik.

5.4 Pengujian Titik Berat

5.4.1 Tujuan

Untuk mengetahui posisi titik berat pada perangkat keras setelah semua perangkat keras terangkai.

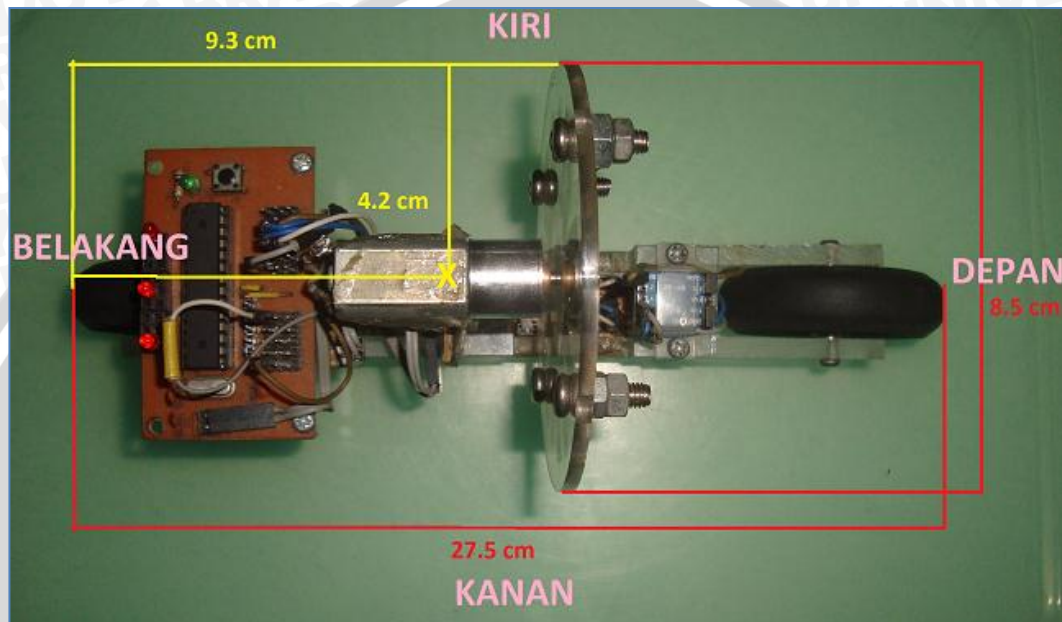
5.4.2 Peralatan yang Digunakan

- Plant miniatur sepeda motor lengkap dengan *accelerometer*, motor DC, Mikrokontroler ATmega8, dan *Driver* motor.
- Benang.
- Penggaris.

5.4.3 Langkah Pengujian

1. Merangkai perangkat keras.
2. Mengantung perangkat keras.
3. Menggeser-geser posisi benang.

Mula-mula miniatur sepeda motor digantung pada seutas benang diantara posisi depan dan belakang selanjutnya benang di geser-geser sehingga berat antara depan dan belakang seimbang, setelah di temukan titik keseimbangan diantara depan dan belakang miniatur sepeda motor digantung diantara kiri dan kanan miniatur sepeda motor sama seperti sebelumnya benang di geser-geser sehingga kiri dan kanan miniatur sepeda motor seimbang.



Gambar 5.8 Titik Keseimbangan Miniatur Sepeda Motor

Gambar 5.8 menunjukkan titik keseimbangan miniatur sepeda motor. Titik keseimbangan miniatur sepeda motor berada pada :

- 4.2 cm dari kiri sisi terluar miniatur sepeda motor
- 9.3 cm dari belakang sisi terluar miniatur sepeda motor

5.5 Pengujian Keseluruhan

5.5.1 Tujuan

Untuk mengetahui kerja dari perangkat keras dan perangkat lunak setelah diintegrasikan dalam sebuah sistem terpadu.

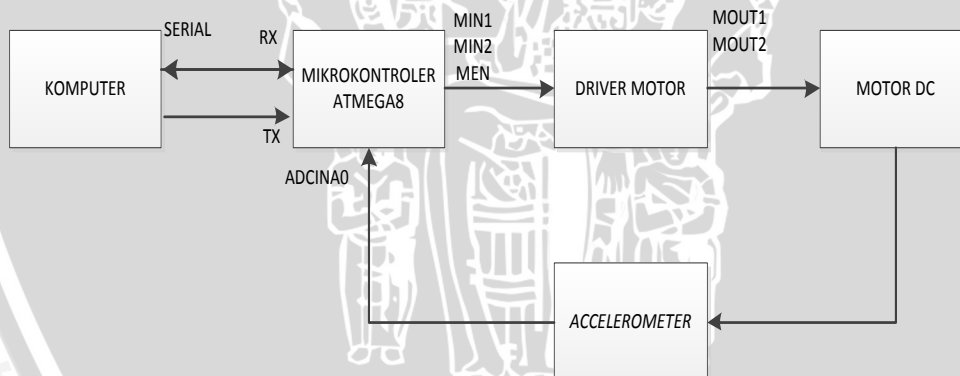
5.5.2 Peralatan yang Digunakan

- Plant miniatur sepeda motor lengkap dengan *accelerometer* dan motor DC
- Mikrokontroler ATmega8
- *Driver* motor

- Serial Port Max 232
- Program *Microsoft Excel*
- *Software CodeVisionAVR Pro v2.03.9.*
- Program *Visual Basic 6* untuk menampilkan performansi sistem.

5.5.3 Langkah Pengujian

1. Memasang alat yang dibutuhkan seperti pada Gambar 5.8.
2. Mengaktifkan semua catu daya.
3. Membuat program untuk pengendalian miniatur sepeda motor dengan kontroler fuzzy pada *Software Code Vision AVR Pro v2.03.9.*
4. Melakukan *make a project* pada software *Software Code Vision AVR Pro v2.03.9*
5. Menegakkan posisi miniatur sepeda motor terhadap posisi pijak, kemudian sistem diaktifkan.
6. Mengamati kinerja dari miniatur sepeda motor dalam menjaga keseimbangan.
7. Merekam data secara *real time* melalui program *Visual Basic 6*.



Gambar 5.9 Blok Diagram Pengujian Sistem Secara Keseluruhan

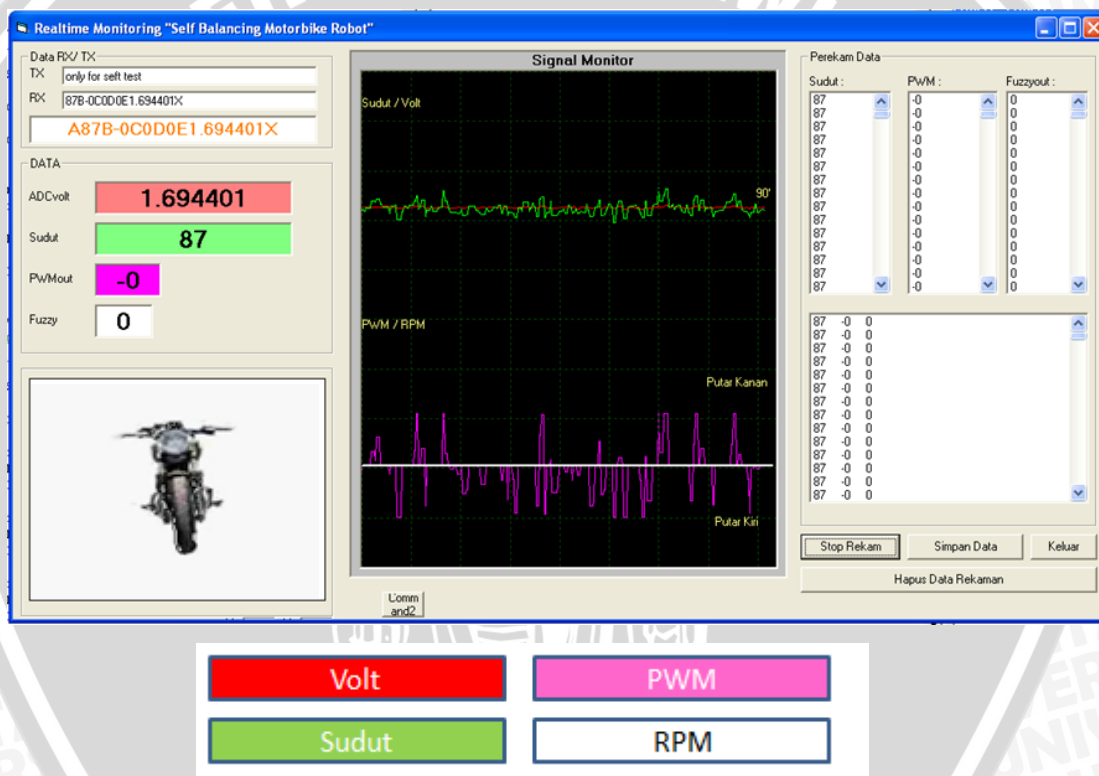
Gambar 5.9 menunjukkan diagram blok pengujian keseluruhan sistem keseimbangan miniatur sepeda motor. Mula-mula miniatur sepeda motor diberdirikan 90^0 kemudian dilepaskan untuk melihat kemampuan sistem penyeimbang dalam menjaga keseimbangan miniatur sepeda motor.

Parameter yang diamati pada skripsi ini adalah lama waktu sistem dapat bertahan pada keadaan setimbang pada posisi 90^0 sebelum *collapse* atau terjatuh. Untuk setiap parameter kontroler pengujian dilakukan hingga miniatur sepeda motor *collapse*.

Pengujian pada saat motor posisi tegak, posisi miring dan posisi jatuh untuk melihat pembacaan *accelerometer*, kinerja motor DC, PWM, dan ADC pada keseluruhan sistem serta melihat kerja *software Visual Basic 6*, sehingga bisa mengoptimalkan kinerja kontroler *fuzzy*.

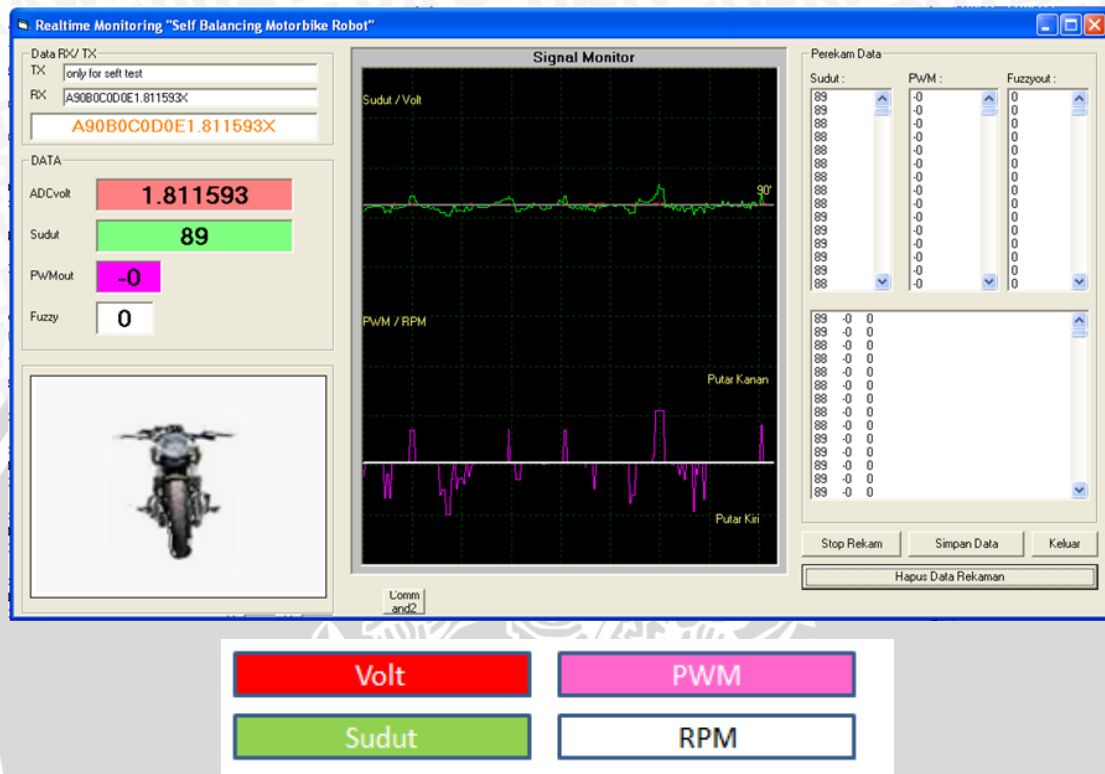
5.5.4 Hasil Pengujian Keseluruhan

- a) Pengujian motor pada saat posisi tegak. Pada Gambar 5.10 dapat dilihat pembacaan respon keseluruhan sistem miniatur sepeda motor pada saat posisi 87° terhadap permukaan pijak. Pembacaan keseluruhan sistem menggunakan menggunakan program *visual basic 6*.



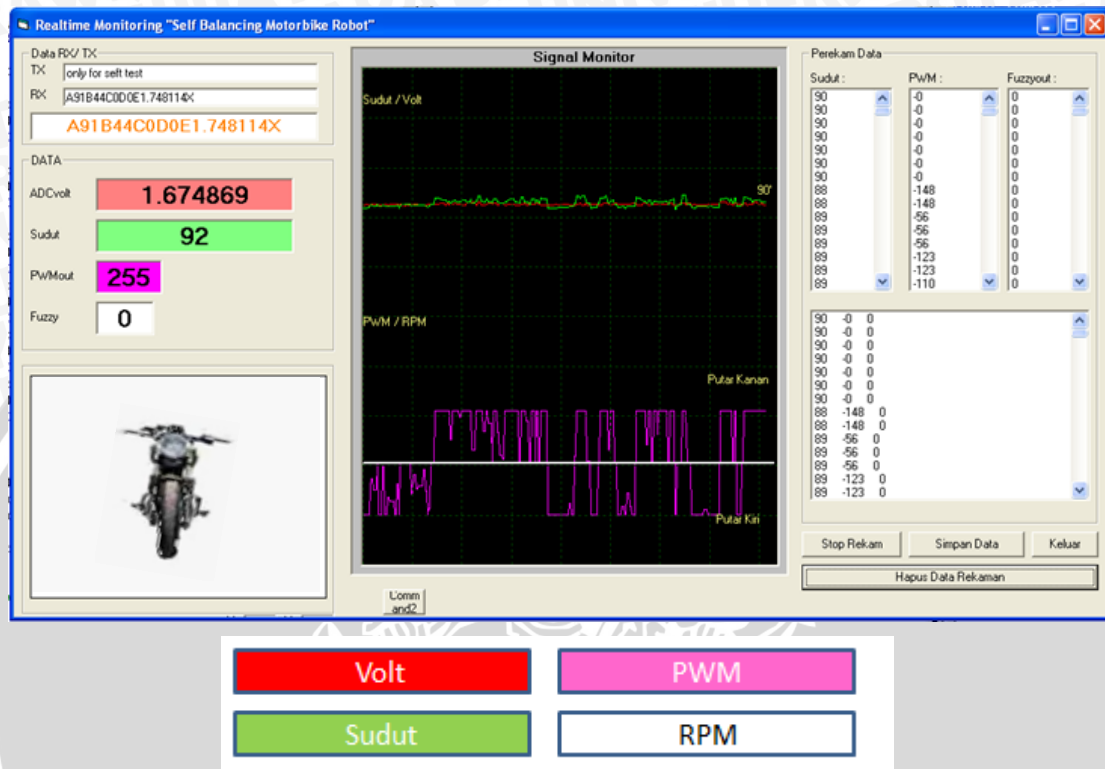
Gambar 5.10 Pengambilan Data Pada Saat Posisi Miniatur Sepeda Motor Tidak Tegak Lurus atau 87° dari Permukaan Pijak

- b) Pengujian motor pada saat posisi tegak. Pada Gambar 5.11 dapat dilihat pembacaan respon keseluruhan sistem miniatur sepeda motor pada saat posisi 89^0 terhadap permukaan pijak. Pembacaan keseluruhan sistem menggunakan menggunakan program *visual basic 6*.



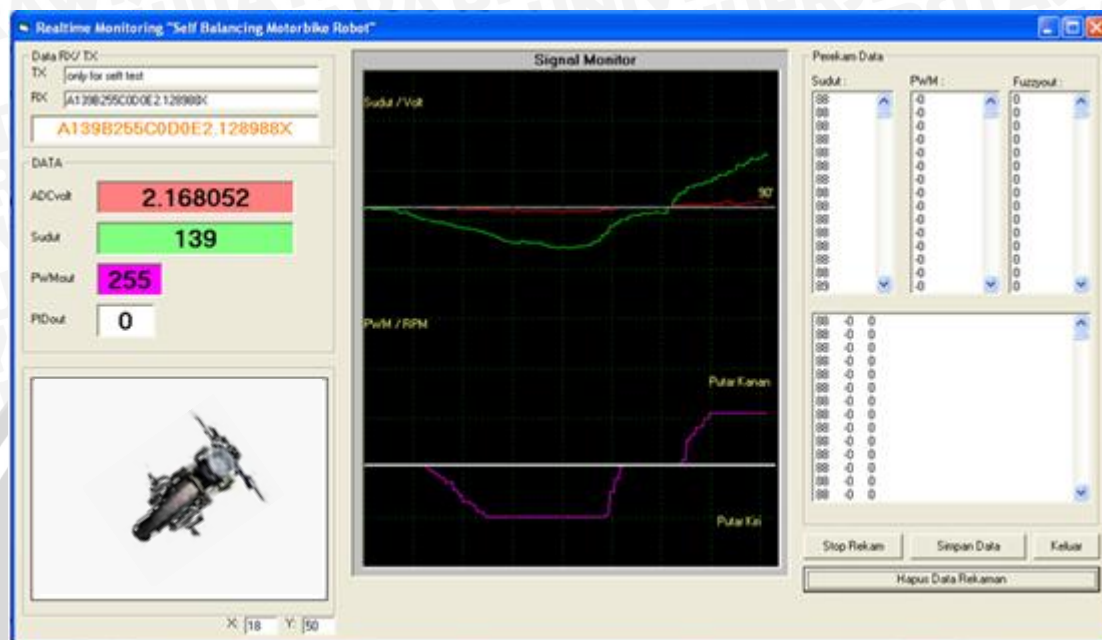
Gambar 5.11 Pengambilan Data Pada Saat Posisi Miniatur Sepeda Motor Tidak Tegak Lurus atau 89^0 dari Permukaan Pijak

- c) Pengujian motor pada saat posisi tegak. Pada Gambar 5.12 dapat dilihat pembacaan respon keseluruhan sistem miniatur sepeda motor pada saat posisi 92^0 terhadap permukaan pijak. Pembacaan keseluruhan sistem menggunakan menggunakan program *visual basic 6*.



Gambar 5.12 Pengambilan Data Pada Saat Posisi Miniatur Sepeda Motor Tidak Tegak Lurus atau 92^0 dari Permukaan Pijak

- d) Pengujian motor pada saat posisi tegak. Pada Gambar 5.13 dapat dilihat pembacaan respon keseluruhan sistem miniatur sepeda motor pada saat posisi 139° terhadap permukaan pijak. Pembacaan keseluruhan sistem menggunakan menggunakan program *visual basic 6*.



Gambar 5.13 Pengambilan Data Pada Saat Posisi Miniatur Sepeda Motor Tidak Tegak Lurus atau 139° dari Permukaan Pijak

Secara keseluruhan dari pengujian ini dapat dilihat bahwa miniatur sepeda motor yang dirancang telah berjalan dengan baik. Proses penyeimbangan dengan bergerak sesuai arah dan kecepatan tertentu dapat berjalan sebagaimana mestinya.

BAB VI PENUTUP

6.1 Kesimpulan

Dari perancangan, pengujian dan pengamatan yang telah dilakukan pada miniatur sepeda motor maka dapat diambil kesimpulan sebagai berikut :

1. Penggunaan *accelerometer* sebagai sensor pembacaan kemiringan sudut sudah bekerja dengan keakuratan yang baik begitu pula dengan kinerja motor DC *gearbox* yang mengatur *disc inertia* sebagai upaya untuk menjaga keseimbangan sudah sesuai harapan. Namun, perancangan mekanik miniatur sepeda motor belumlah sempurna, sehingga belum tercapainya performansi sistem sesuai dengan yang diinginkan.
2. Dengan menggunakan teorema kontroler *fuzzy* yang ditanamkan pada mikrokontroler ATmega8, menggunakan *software Code Vision AVR Pro v2.03.9*, sistem dapat mengambil keputusan untuk berputar searah putaran jarum jam atau berkebalikan arah jarum jam dengan kecepatan tertentu dari motor DC sebagai upaya untuk menyeimbangkan miniatur sepeda motor. Dalam penelitian ini sistem dapat mempertahankan keseimbangan selama kurang lebih 10 detik.

6.2 Saran

Dalam perancangan dan pembuatan alat ini masih terdapat kelemahan. Untuk memperbaiki kinerja alat dan pengembangan lebih lanjut disarankan:

1. Penyempurnaan konstruksi mekanik pada sistem miniatur sepeda motor agar dicari titik keseimbangannya sehingga motor dapat berdiri tegak sebelum adanya aksi pengontrolan. Penyempurnaan pada *disc inertia* untuk dibuat lebih simetris, hal ini untuk mengurangi *noise* dan *disturbance* saat sistem sedang berjalan.
2. *Accelerometer* yang digunakan sebagai pembacaan sudut dapat diganti dengan menggunakan *gyroscope*. Dapat juga menggunakan kombinasi antara *accelerometer* dan *gyroscope* agar pembacaan sudut lebih baik dan lebih tepat.
3. Dilakukan metode pengontrolan selain menggunakan kontroler *fuzzy*.

4. Menambahkan data pengujian yang digunakan untuk mendapatkan rata-rata waktu pengontrolan pada pengujian keseluruhan.



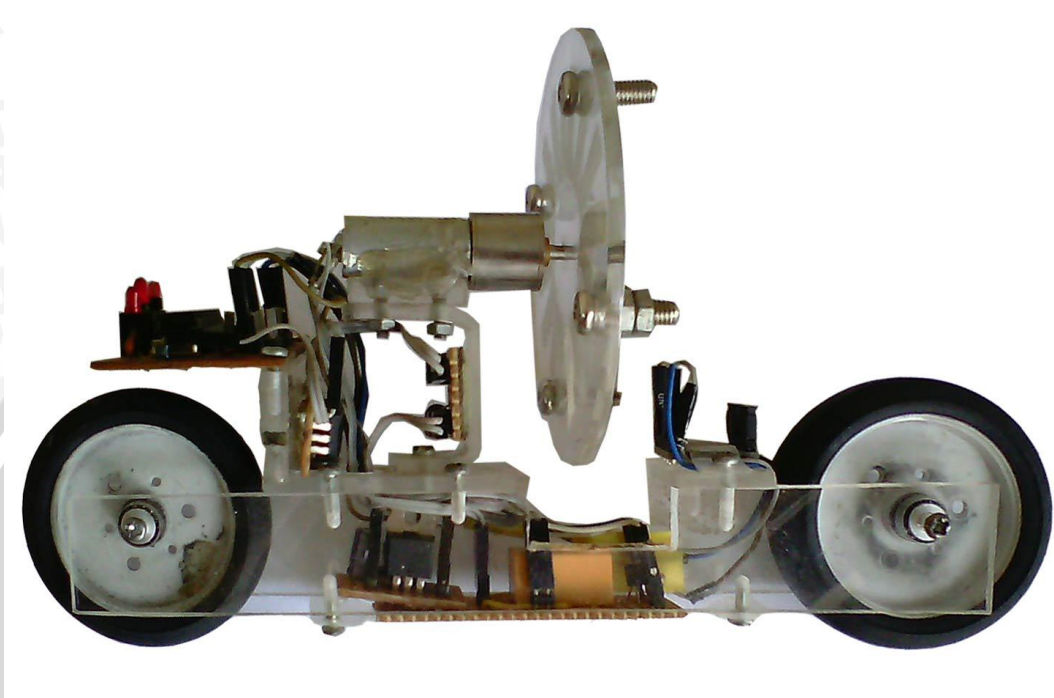
DAFTAR PUSTAKA

- ATMEL. 2007. *ATMEGA8535/ATMEGA8535L, 8-bit AVR Microcontroller with 8 Kbytes in System Programable Flash*.
- Innovative Electronics. *EMS 5A H-Bridge Datasheet*.
- Jantzen, Jan. 1998. *Tuning of Fuzzy PID Controller*, Technical University of Denmark, Department of Automation. Denmark.
- Ogata, K. 1997. *Terjemahan Teknik Kontrol Automatik Jilid 1*. Jakarta. Penerbit Erlangga.
- Ogata, K. 1997. *Terjemahan Teknik Kontrol Automatik Jilid 2*. Jakarta. Penerbit Erlangga.
- Putra, Al Ridho W. 2011. *Perancangan dan pembuatan rotary inverted pendulum dengan menggunakan kontrol logika fuzzy*, Malang. Fakultas Teknik Universitas Brawijaya.
- Rahman, Ari. 2010. *Desain dan implementasi control logika fuzzy untuk pengendalian kestabilan pendulum terbalik*, Malang. Fakultas Teknik Universitas Brawijaya.
- Soemarwanto. 1997. *Dasar Konversi Energi Elektrik Jilid II*. Malang. Fakultas Teknik Universitas Brawijaya .
- Stephani, Herlina. 2010. *Pengendalian Optimal Furuta Pendulum*, Surabaya. Fakultas Matematika dan Ilmu Pengetahuan Alam Institut Teknologi Sepuluh Nopember.
- Sulistiono, A. 2010. *Pengenalan : Pulse Width Modulation*. <http://www.ariststy.cz/cc/2010/02/pulse-width-modulation-pwm-pengenalan.html>
- Texas Instrument Inc. 2002. *TMS320F2812 Digital Signal Processor Implementation tutorial*.

LAMPIRAN I

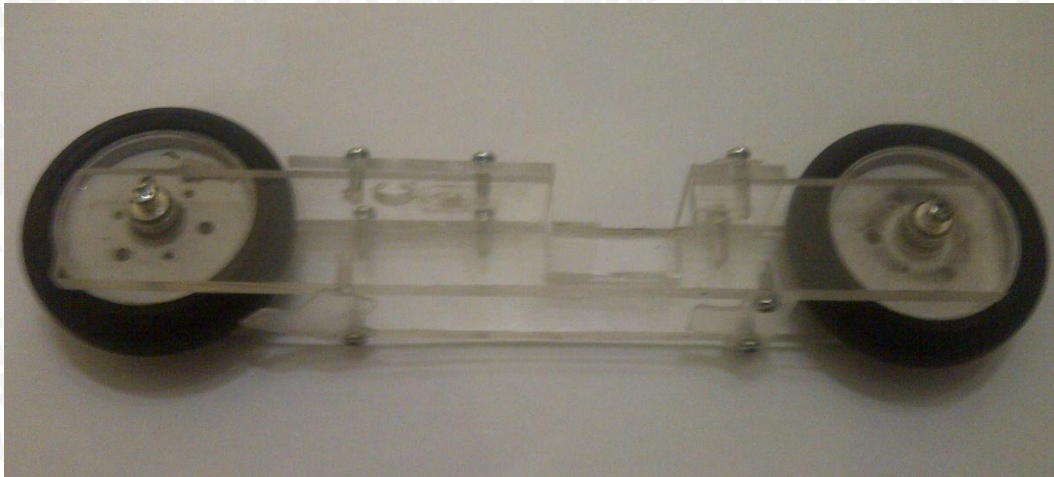


DAFTAR GAMBAR ALAT

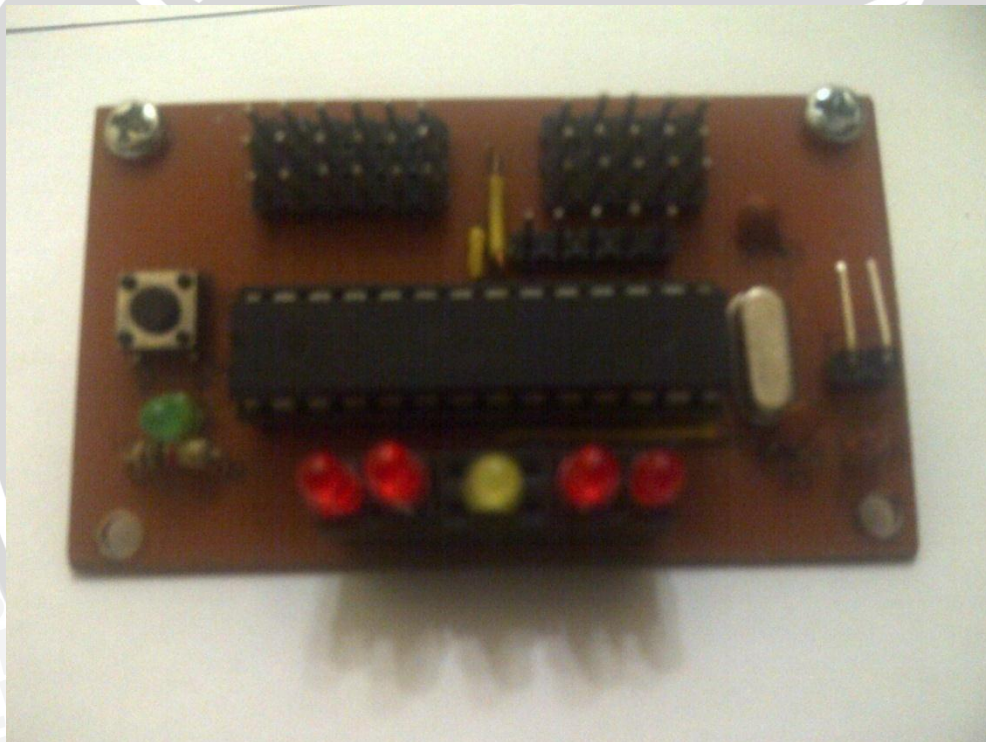


Gambar 1. Gambar Alat Tampak Samping





Gambar 2. Kerangka dan Roda Alat



Gambar 3. Modul Mikrokontroler ATmega8

LAMPIRAN II



LISTING PROGRAM

/******

This program was produced by the
CodeWizardAVR V2.03.9 Standard
Automatic Program Generator

© Copyright 1998-2008 Pavel Haiduc, HP InfoTech s.r.l.

<http://www.hpinfotech.com>

Project :

Version :

Date : 5/14/2012

Author : ----

Company :

Comments:

Chip type : ATmega8

Program type : Application

AVR Core Clock frequency: 12.000000 MHz

Memory model : Small

External RAM size : 0

Data Stack size : 256

*****/

#include <mega8.h>

#include <bike.h>

//Alphanumeric LCD Module functions

##asm

```
// .equ __lcd_port=0x12 ;PORTD
//endasm
#include <lcd.h>
#include <stdio.h>
#include <delay.h>
#include <stdlib.h>
#include <math.h>
#define ADC_VREF_TYPE 0x40
float VoutX, CenterX;
int cnt,x,i;
float sudut;
float error, delta_error;
float angle[101],Sangle,X_VAL;
float v_rpm, rpm_act, rpm_sp;
int cnt_enc;
char MF,d_MF;
float FZ_OUT, PWM;
float e,ed;
float FZ_e, FZ_ed;
float N, Z, NZ, PZ, P;
float eN, eZ, eNZ, ePZ, eP;
float edN, edZ, edNZ, edPZ, edP;
float cN, cZ, cNZ, cPZ, cP;
float vxN,vN,vZ,vNZ,vPZ,vP,vxp;
float mvxN,mvN,mvZ,mvNZ,mvPZ,mvP,mvxp;
float MO_N,MO_NZ,MO_Z,MO_PZ,MO_P,MO_OUT;
float FA,FB,FC,FD,F1,F2,F3,F4,FMINX,FMAXX;
```

```

char mem_e, mem_ed, mem_fz;

// Read the AD conversion result

unsigned int read_adc(unsigned char adc_input)
{
    ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);

    // Delay needed for the stabilization of the ADC input voltage
    delay_us(10);

    // Start the AD conversion
    ADCSRA|=0x40;

    // Wait for the AD conversion to complete
    while ((ADCSRA & 0x10)==0);

    ADCSRA|=0x10;

    return ADCW;
}

// External Interrupt 0 service routine
interrupt [EXT_INT0] void ext_int0_isr(void)
{
    // counter encoder
    cnt_enc++;
}

void baca_adc()
{
    VoutX = (float)read_adc(0);
    v_rpm = (float)read_adc(1);
}

// Timer 2 overflow interrupt service routine
interrupt [TIM2_OVF] void timer2_ovf_isr(void)
{

```

```

cnt++;
if(cnt<120)
{
    if(x==1)
    {
        OCR1A=abs(PWM);
    }
}

if(cnt>24000)    //reset counter 1 detik
{
    //rpm_act=(cnt_enc/16)*60; //RPM tidak dipakai
    rpm_act=cnt_enc;    //ini yang dipakai Pulse per second yg lebih
    presisi
    cnt_enc=0;
    cnt=0;
}
}

void realtime_sudut()
{
    baca_adc();    //iterasi pembacaan input ADC / sensor
    accelero

    for(i=0;i<100;i++)    //Filter Rata-rata 100 data
    {
        baca_adc();
        CenterX=CenterX+VoutX;
    }

    CenterX=(CenterX/100);
    for(i=0;i<100;i++)    //Filter Rata-rata 100 data
    {

```

```

        if(CenterX<=377)           //nilai 357 = nilai ADC accelero kondisi
            tegak lurus 90'
        {
            sudut((((CenterX-180))*(0.564972))*0.9;
        }
        else
        {
            sudut=90+((((CenterX-377))*(0.584795))*0.9);
        }
        angle[i]=sudut;
    }
    for(i=0;i<100;i++)
    {
        Sangle=Sangle+angle[i];
    }
    Sangle=(Sangle/100)-3;
}

void indikator(float data)
{
    if(data>95 && data<180)
    {
        PORTD.0=1;
    }
    else{PORTD.0=0;}

    //if(data>91 && data<95) {PORTD.2=1;} else{PORTD.2=0;}

    if(data>89 && data<91) {PORTD.4=1;} else{PORTD.4=0;}

    if(data<89&& data>85) {PORTD.6=1;} else{PORTD.6=0;}

    if(data<85 && data>10) {PORTB.0=1;} else{PORTB.0=0;}

```

```
}  
// Declare your global variables here  
float triagle(float error, float min, float mid, float max)  
{  
    float range, value;  
    if(error>=mid)  
    {  
        range=max-mid;  
        value=1-((1/range)*(error-mid));  
    }  
    if(error<=mid)  
    {  
        range=min-mid;  
        value=((1/range)*(min-error));  
    }  
    if(value>1){value=1;}  
    if(value<0){value=0;}  
    return value;  
}  
float slash(float error, float min, float mid, float max)  
{  
    float range, value;  
    if(error>=mid)  
    {  
        value=1;  
    }  
    if(error<=mid)  
    {
```

```

    range=min-mid;
    value=((1/range)*(min-error));
}
if(value>1){value=1;}
if(value<0){value=0;}
return value;
}

float bslash(float error, float min, float mid, float max)
{
    float range, value;

    if(error>=mid)
    {
        range=max-mid;
        value=1-((1/range)*(error-mid));
    }
    if(error<=mid)
    {
        value=1;
    }
    if(value>1){value=1;}
    if(value<0){value=0;}
    return value;
}

float motor_out(float FZ_val, float min, float max)
{
    float value;
    float result;

```

```

value = max-min;
result=min+(FZ_val*(value/FZ_val));
return result;
}

void main(void)
{
    setup();

    // ADC initialization
    ADMUX=ADC_VREF_TYPE & 0xff;
    ADCSRA=0x84;
    #asm("sei")

    while (1)
    {
        //=====//

        //proses ini digunakan untuk 'PERBAIKAN ALAT'

        if(x==0)        //Proses Kalibrasi mencari nilai ADC/Sensor
        Accelerometer/tegak lurus/90'
        {
            //dilakukakn hanya saat awal program.

            for(i=0;i<100;i++)
            {
                baca_adc();
                CenterX=CenterX+VoutX;
            }

            CenterX=CenterX/100;

            X_VAL=CenterX;

            x=1;

            printf("Loading data..");

```

```

delay_ms(50);
}

//=====================================================PROSES PEMBACAAN SUDUT//=====================================================//

realtime_sudut();           //mendapatkan nilai sudut secara realtime
"Sangle"

indikator(Sangle);          //tampilan indikator sudut 0 - 180'
menggunakan LED

//=====================================================END//=====================================================//

//=====================================================FUZZY / Set Kecepatan Motor//=====================================================//

// xn      N      NZ      Z      PZ      P      xp
// $$$$$$$$$$$$$$      #      $$      #      $$$$$$$$$$$$$$
//          $ # #      $ $      # # $
//          #$      #$      $#      $#
//          #      $ $      #      #      $ $      #
//          #      $      ##      $      #
// -100      -50      -25      0      25      50      100
// vxn      vn      vnz      vz      vpz      vp      vxp

error=Sangle-90;
vx=-100;
vn=-20;
vnz=-10;
vz=0;
vpz=10;
vp=20;
vxp=100;

```

mvxn=-255;

mvn=-100;

mvnz=-50;

mvz=0;

mvpz=50;

mvp=100;

mvxp=255;

eN=bslash(error,vxn,vn,vnz);

eNZ=triagle(error,vn,vnz,vz);

eZ=triagle(error,vnz,vz,vpz);

ePZ=triagle(error,vz,vpz,vp);

eP=slash(error,vpz,vp,vxp);

delta_error=fabs(error-delta_error);

edN=bslash(delta_error,vxn,vn,vnz);

edNZ=triagle(delta_error,vn,vnz,vz);

edZ=triagle(delta_error,vnz,vz,vpz);

edPZ=triagle(delta_error,vz,vpz,vp);

edP=slash(delta_error,vpz,vp,vxp);

//BERSIHKAN DATA COA

cN=0; cNZ=0; cZ=0; cPZ=0; cP=0;

//1 eN x edN

//isi N dengan nilai terkecil

//isi COA N dengan nilai terbesar

if(eN<edN){N=eN;} if(eN>edN){N=edN;} if(N>cN){cN=N;}

//2 eN x edNZ

//isi N dengan nilai terkecil //isi COA N dengan nilai terbesar

if(eN<edNZ){N=eN;} if(eN>edNZ){N=edNZ;} if(N>cN){cN=N;}

//3 eN x edZ

//isi N dengan nilai terkecil //isi COA N dengan nilai terbesar

if(eN<edZ){N=eN;} if(eN>edZ){N=edZ;} if(N>cN){cN=N;}

//4 eN x edPZ

//isi N dengan nilai terkecil //isi COA NZ dengan nilai terbesar

if(eN<edPZ){NZ=eN;} if(eN>edPZ){NZ=edPZ;} if(NZ>cNZ){cNZ=NZ;}

//5 eN x edP

//isi N dengan nilai terkecil //isi COA NZ dengan nilai terbesar

if(eN<edP){NZ=eN;} if(eN>edP){NZ=edP;} if(NZ>cNZ){cNZ=NZ;}

//---->

//6 eNZ x edN

//isi NZ dengan nilai terkecil //isi COA N dengan nilai terbesar

if(eNZ<edN){N=eNZ;} if(eNZ>edN){N=edN;} if(N>cN){cN=N;}

//7 eNZ x edNZ

//isi NZ dengan nilai terkecil //isi COA N dengan nilai terbesar

if(eNZ<edNZ){N=eNZ;} if(eNZ>edNZ){N=edNZ;} if(N>cN){cN=N;}

//8 eNZ x edZ

//isi NZ dengan nilai terkecil //isi COA N dengan nilai terbesar
 if(eNZ<edZ){NZ=eNZ;} if(eNZ>edZ){NZ=edZ;} if(NZ>cNZ){cNZ=NZ;}

//9 eNZ x edPZ

//isi NZ dengan nilai terkecil //isi COA NZ dengan nilai terbesar
 if(eNZ<edPZ){Z=eNZ;} if(eNZ>edPZ){Z=edPZ;} if(Z>cZ){cZ=Z;}

//10 eNZ x edP

//isi NZ dengan nilai terkecil //isi COA NZ dengan nilai terbesar
 if(eNZ<edP){PZ=eNZ;} if(eNZ>edP){PZ=edP;} if(PZ>cPZ){cPZ=PZ;}

//---->

//11 eZ x edN

//isi N dengan nilai terkecil //isi COA N dengan nilai terbesar
 if(eZ<edN){N=eZ;} if(eZ>edN){N=edN;} if(N>cN){cN=N;}

//12 eZ x edNZ

//isi NZ dengan nilai terkecil //isi COA NZ dengan nilai terbesar
 if(eZ<edNZ){NZ=eZ;} if(eZ>edNZ){NZ=edNZ;} if(NZ>cNZ){cNZ=NZ;}

//13 eZ x edZ

//isi Z dengan nilai terkecil //isi COA Z dengan nilai terbesar
 if(eZ<edZ){Z=eZ;} if(eZ>edZ){Z=edZ;} if(Z>cZ){cZ=Z;}

//14 eZ x edPZ

//isi PZ dengan nilai terkecil //isi COA PZ dengan nilai terbesar
 if(eZ<edPZ){PZ=eZ;} if(eZ>edPZ){PZ=edPZ;} if(PZ>cPZ){cPZ=PZ;}

//15 eZ x edP

//isi P dengan nilai terkecil //isi COA P dengan nilai terbesar

if(eZ<edP){P=eZ;} if(eZ>edP){P=edP;} if(P>cP){cP=P;}

//---->

//16 ePZ x edN

//isi NZ dengan nilai terkecil //isi COA NZ dengan nilai terbesar

if(ePZ<edN){NZ=ePZ;} if(ePZ>edN){NZ=edN;} if(NZ>cNZ){cNZ=NZ;}

//17 ePZ x edNZ

//isi Z dengan nilai terkecil //isi COA Z dengan nilai terbesar

if(ePZ<edNZ){Z=ePZ;} if(ePZ>edNZ){Z=edNZ;} if(Z>cZ){cZ=Z;}

//18 ePZ x edZ

//isi PZ dengan nilai terkecil //isi COA PZ dengan nilai terbesar

if(ePZ<edZ){PZ=ePZ;} if(ePZ>edZ){PZ=edZ;} if(PZ>cPZ){cPZ=PZ;}

//19 ePZ x edPZ

//isi P dengan nilai terkecil //isi COA P dengan nilai terbesar

if(ePZ<edPZ){P=ePZ;} if(ePZ>edPZ){P=edPZ;} if(P>cP){cP=P;}

//20 ePZ x edP

//isi P dengan nilai terkecil //isi COA P dengan nilai terbesar

if(ePZ<edP){P=ePZ;} if(ePZ>edP){P=edP;} if(P>cP){cP=P;}

//---->

//21 eP x edN

//isi PZ dengan nilai terkecil //isi COA PZ dengan nilai terbesar

```
if(eP<edN){PZ=eP;} if(eP>edN){PZ=edN;} if(PZ>cPZ){cPZ=PZ;}
```

```
//22 eP x edNZ
```

```
//isi PZ dengan nilai terkecil //isi COA PZ dengan nilai terbesar
```

```
if(eP<edNZ){PZ=eP;} if(eP>edNZ){PZ=edNZ;} if(PZ>cPZ){cPZ=PZ;}
```

```
//23 eP x edZ
```

```
//isi P dengan nilai terkecil //isi COA PZ dengan nilai terbesar
```

```
if(eP<edZ){P=eP;} if(eP>edZ){P=edZ;} if(P>cP){cP=P;}
```

```
//24 eP x edPZ
```

```
//isi P dengan nilai terkecil //isi COA P dengan nilai terbesar
```

```
if(eP<edPZ){P=eP;} if(eP>edPZ){P=edPZ;} if(P>cP){cP=P;}
```

```
//25 eP x edP
```

```
//isi P dengan nilai terkecil //isi COA P dengan nilai terbesar
```

```
if(eP<edP){P=eP;} if(eP>edP){P=edP;} if(P>cP){cP=P;}
```

```
//---defuzifikasi COA
```

```
MO_OUT=0;
```

```
MO_OUT=((cN*mvn)+(cNZ*mvnz)+(cZ*mvz)+(cPZ*mvpz)+(cP*mvp))/(cN  
+cNZ+cZ+cPZ+cP);
```

```
MO_OUT=FA+FB+FC+FD;
```

```
PWM=255*MO_OUT;
```

```
if(PWM==0)
```

```
{
```

```
PWM=30;
```

```
}
```

```
if(PWM>255 || PWM<-255)
```

```

{
    PWM=255;
}

if(Sangle>100){PWM=255;}

if(Sangle<80){PWM=255;}

//=====================================================END=====//

//=====DISK BALANCING SYSTEM TURN LEFT OR RIGHT=====//

    if(Sangle<90 &&
    Sangle>87){PORTB.4=0;PORTB.5=1;}if(Sangle<90){PORTB.4=1;PORTB.5
    =0;} //LEFT

    if(Sangle>90 &&
    Sangle<93){PORTB.4=1;PORTB.5=0;}if(Sangle>90){PORTB.4=0;PORTB.5
    =1;} //RIGHT

    if(PORTB.5==0){rpm_act=rpm_act*1;}

    if(PORTB.5==1){rpm_act=rpm_act*-1;}

    //=====KIRIM DATA KE KOMPUTER=====//
    if(error>=0)
    {

        printf("A%.0fB%fC%.0fD%.0fE%fX",Sangle,PWM,rpm_act,rpm_sp,Vout
        X*0.004883);

    }

    if(error<0)
    {

        printf("A%.0fB-
        %fC%.0fD%.0fE%fX",Sangle,PWM,rpm_act,rpm_sp,VoutX*0.004883);

    }

    //=====//

    delay_ms(20);

};

}

```

LAMPIRAN III

