

**KONTROL KECEPATAN PEMUTAR GERABAH DENGAN
KONTROLER PID MENGGUNAKAN MIKROKONTROLER
ATMEGA8535**

SKRIPSI

*Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik*



Disusun oleh:
KHASBULLAH HUDA
NIM. 0710630045-63

**KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS BRAWIJAYA
FAKULTAS TEKNIK
MALANG**

2012

LEMBAR PERSETUJUAN

KONTROL KECEPATAN PEMUTAR GERABAH DENGAN KONTROLER PID MENGGUNAKAN MIKROKONTROLER ATMEGA8535

SKRIPSI KONSENTRASI TEKNIK KONTROL

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik



Disusun oleh:
KHASBULLAH HUDA
NIM. 0710630045-63

Telah diperiksa dan disetujui oleh:

Pembimbing I

Pembimbing II

Retnowati, Ir., MT.
NIP. 19511224 198203 2 001

Tri Nurwati, ST., MT.
NIP. 19790615 200812 2 003



LEMBAR PENGESAHAN

**KONTROL KECEPATAN PEMUTAR GERABAH
DENGAN KONTROLER PID MENGGUNAKAN
MIKROKONTROLER ATMEGA8535**

SKRIPSI

JURUSAN TEKNIK ELEKTRO

Diajukan untuk Memenuhi Sebagian Persyaratan
Memperoleh Gelar Sarjana Teknik

Disusun oleh:

KHASBULLAH HUDA

NIM. 0710630045-63

Skripsi ini telah diuji dan dinyatakan lulus pada
tanggal 31 Januari 2012

DOSEN PENGUJI

Goegoes Dwi Nusantara, ST., MT.
NIP. 19711013 200604 1 001

M Aziz Muslim.,ST.,MT.,Ph D
NIP.19741203 200012 1 001

Ir. Purwanto, MT.
NIP. 19540424 198601 1 001

Mengetahui
Ketua Jurusan Teknik Elektro

DR. Ir. Sholeh Hadi Pramono, MS.
NIP. 19580728 198701 1 001

KATA PENGANTAR

Alhamdulillah, segenap puji dan syukur penulis ucapkan kehadirat Allah SWT yang telah melimpahkan rahmat, hidayah, ridho, nikmat dan karunia-Nya sehingga penulis dapat menyelesaikan tugas akhir ini dengan judul “KONTROL KECEPATAN PEMUTAR GERABAH DENGAN KONTROLER PID MENGGUNAKAN MIKROKONTROLER ATMEGA8535”.

Dengan kesungguhan dan rasa rendah hati, penulis mengucapkan terimakasih kepada Ibu Retnowati, Ir., MT., dan Ibu Tri Nurwati, ST., MT., sebagai dosen pembimbing atas saran, konsultasi, motivasi, kesabaran dan waktu yang telah dicurahkan dalam pengerjaan hingga penyelesaian skripsi ini. Tidak terkecuali, ucapa nterimakasih penulis ucapkan kepada:

1. Orang - orang rumah. Mamah, terimakasih atas semuanya. Terlalu banyak kebaikan mamah sehingga aku tidak bias menulisnya. canda, tawa, marah, tangis, dan haru, menghiasi hari-hari mamah dalam membesarkan aku selama ini. Semoga setelah ini Cuma ada canda, tawa, dan haru ya mah?. Papah, terimakasih atas kedisiplinan dan rasa tanggungjawab yang diajarkan kepadaku. Didikan dari papah sangat membantuku dalam menyelesaikan pendidikan di bangku sekolah hingga bangku kuliah, dan akan terus berguna pah. say no to mental tempe, thats right pah. Adek, terimakasih atas semuanya, udah bantu aku ngelewatin masa – masa galau. kuliah yang bener ya dek, lulus tepat waktu, jangan molor kayak aku.
2. Bapak Sholeh Hadi Pramono, Dr., Ir., Ms. Selaku Ketua Jurusan Teknik Elektro Universitas Brawijaya
3. Bapak Muh. Aziz Muslim, ST., MT., Ph.D. selaku Sekretaris Jurusan Teknik Elektro Universitas Brawijaya
4. Bapak Ir. Purwanto MT. selaku Ketua Kelompok Dosen dan Keahlian Teknik Kontrol yang telah membantu dalam menyelesaikan proses penyelesaian skripsi ini mulai seminar proposal, hasil dan ujian yang dilakukan penulis
5. Bapak Goegoes D. N. ST., MT. selaku dosen pembimbing akademik

6. Bapak, Ibu dosen serta segenap staf dan karyawan Jurusan Teknik Elektro baik secara langsung maupun tidak langsung yang telah banyak membantu dalam menyelesaikan skripsi ini.
7. Nia, makasih uda selalu ada dan selalu menemani dalam pengerjaan skripsi ini. Kamu selalu ada disetiap *moment* yang aku lewati, dan akan terus demikian adanya. Ini semua akan menjadi kenangan yang manis.
8. Afri, Subekti, Keprek, Afriono, rido', wahyu, yoga, vandra, eres, danwito '09. Terimakasih sekali, kalian sudah meluangkan waktu dan tenaga untuk membantu aku. Sungguh kawan yang super, super sekali.
9. Jemblung, Enos, Yoneth, dan Oji. Terimakasih, kalian teman yang hebat.
10. Anak – anak kos Jl.Kertorahayu No.67. Jilid2 :Faisal, ridlo, ikbal, indro, jaya, barok, anton, ghariza, reza, bli alias rizal,radit, zuhdi, randi, udin, dan mas rudi. Jilid1 : kang iman, kang dadang, surya, agung, dan widi,. Jilid 1 danJilid 2, tidak ada perubahan yang signifikan. Sungguh kosan yang alay, yang berisikan anak – anak galau. Mau dibawa kemana bangsa ini jika penerus bangsa ini adalah anak – anak galau dan alay seperti kalian. *just kidding*.
11. Temen-temen Core terimakasih atas dukungan dan informasi yang diberikan
12. Sahabat - sahabat UNITANTRI, terimakasih sudah mengajarkan *skill* organisasi.
13. Semua pihak yang tidak dapat saya sebutkan satu persatu. Terimakasih,

Penulis menyadari bahwa yang tersusun dalam skripsi ini masih banyak kekurangan karena keterbatasan materi dan pengetahuan yang dimiliki.Karena itu kritik dan saran sangat diharapkan demi kesempurnaan penyusunan skripsi ini. Akhir kata, penulis mengharapkan semoga yang terdapat dalam skripsi ini bermanfaat untuk kita semua.

Malang, 7 Februari 2012

Penulis

ABSTRAK

Pemutar gerabah merupakan alat yang digunakan untuk membentuk adonan tanah liat menjadi bentuk - bentuk yang simetris seperti bulat, silindris, dan bervariasi. Skripsi ini menjelaskan tentang pengontrolan kecepatan pemutar gerabah yang diputar dengan motor DC. Sebagai pengendali utama digunakan Mikrokontroler dan implementasi kontroler PID sebagai pengendali kecepatan pemutar gerabah. Pengujian pada pemutar gerabah dilakukan dengan mengatur kecepatan pemutar gerabah dengan menentukan nilai tegangan dan penggunaan sensor rotary encoder sebagai pembaca putaran pada pemutar gerabah. Kemudian pemutar gerabah akan diberi beban dan mikrokontroler akan mengatur kecepatan pemutar gerabah agar tetap pada kecepatan yang diinginkan.

Kata kunci: Pengontrolan, Pemutar Gerabah, Mikrokontroler, PID.



DAFTAR ISI

KATA PENGANTAR	ii
DAFTAR ISI	iv
DAFTAR TABEL	vii
DAFTAR GAMBAR	viii
ABSTRAK.....	x
 BAB I PENDAHULUAN	 1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	1
1.3. Batasan Masalah.....	2
1.4. Tujuan.....	2
1.5. Sistematika Pembahasan.....	2
 BAB II TINJAUAN PUSTAKA	 4
2.1. Studi Literatur Gerabah	4
2.1.1 Sekilas Tentang Proses Pembuatan Gerabah.....	5
2.2. Studi Teknik Pengontrolan	6
2.2.1 Kontroler PID (<i>Proportional Integral Derivative</i>).....	6

2.2.1.1 Kontroler Proporsional.....	6
2.2.1.2 Kontroler Integral.....	7
2.2.1.3 Kontroler Differensial.....	8
2.2.1.4 Kontroler Proporsional Integral (PI).....	9
2.2.1.5 Kontroler Proporsional Diferensial (PD).....	10
2.2.1.6 Kontroler Proporsional Integral Differensial (PID).....	10
2.2.2 Mikrokontroler ATmega8535.....	11
2.2.2.1 Arsitektur ATmega8535.....	12
2.2.2.2 Konfigurasi Pin ATmega8535.....	13
2.2.2.3 Peta Memori.....	14
2.3 Studi Sensor.....	17
2.4 Motor DC.....	18
2.5 PWM (<i>Pulse Width Modulation</i>).....	19

BAB III	METODOLOGI PENELITIAN	21
3.1	Studi Literatur.....	21
3.2	Perancangan Alat.....	21
3.3	Pembuatan Alat.....	21
3.3.1	Pembuatan Pemutar Gerabah.....	22
3.3.2	Pembuatan Perangkat Keras.....	22
3.3.3	Pembuatan Perangkat Lunak.....	22
3.4	Pengujian Alat.....	22
3.5	Pengambilan Kesimpulan dan Saran.....	23



BAB IV PERANCANGAN DAN PEMBUATAN ALAT..... 24

4.1 Perancangan Sistem..... 24

4.2 Perancangan Perangkat Keras..... 25

4.2.1 Rangkaian Catu Daya Sistem.....25

4.2.2 Sensor *Rotary Encoder* 26

4.2.3 Rangkaian *Driver Motor*..... 27

4.2.4 Rangkaian Mikrokontroler.....30

4.3 Perancangan Bentuk Mekanik Pemutar Gerabah..... 32

4.4 Perancangan sistem logika *PID* serta implementasinya pada mikrokontroler..... 32

BAB V PENGUJIAN DAN ANALISIS.....39

5.1 Pengujian Catu Daya Sistem.....39

5.2 Pengujian Sensor *Rotary Encoder*.....41

5.3 Pengujian *Driver Motor*.....42

5.4 Pengujian rangkaian mikrokontroler ATmega8535.....43

5.5 Pengujian Keseluruhan Sistem.....45

5.5.1 Peralatan pengujian.....45

5.5.2 Prosedur pengujian.....46

5.5.3 Hasil Pengujian.....46

BAB VI KESIMPULAN DAN SARAN.....57

6.1 Kesimpulan.....57

6.2 Saran.....58

DAFTAR PUSTAKA59

LAMPIRAN 1

Foto Alat

LAMPIRAN 2

Gambar Skematik Alat

LAMPIRAN 3

Listing Program

LAMPIRAN 4

Datasheet



DAFTAR GAMBAR

Gambar 2.1 Diagram blok kontroler proporsional.....	7
Gambar 2.2 Blok diagram kontroler integral.....	8
Gambar 2.3 Blok diagram kontroler differensial.....	9
Gambar 2.4 Diagram blok kontroler PID.....	11
Gambar 2.5 Hubungan fungsi waktu antara sinyal keluaran dan sinyal masukan kontroler PID.....	11
Gambar 2.6 Diagram Blok Fungsional ATmega8535.....	12
Gambar 2.7 Konfigurasi Pin ATmega8535.....	13
Gambar 2.8. Konfigurasi Memori Data ATmega8535.....	15
Gambar 2.9. Memori Program ATmega8535.....	15
Gambar 2.10. Status Register ATmega8535.....	16
Gambar 2.11 Bagan Skematik Sensor <i>Rotary Encoder</i>	17
Gambar 2.13 Motor DC.....	19
Gambar 2.14 Sinyal PWM Secara Umum.....	20
Gambar 4.1. Blok Diagram Sistem.....	24
Gambar 4.2. Rangkaian Catu Daya untuk mikrokontroler.....	25
Gambar 4.3. Rangkaian Catu Daya <i>Coil Rilay</i>	26
Gambar 4.4. Rangkaian Sensor <i>Rotary Encoder</i>	27
Gambar 4.5. Letak <i>Optical IC</i> Pada Piringan Berpola.....	27
Gambar 4.6 Rangkaian <i>driver</i> motor.....	28

Gambar 4.7. Rangkaian Pemisah *Optocoupler* Untuk Coil Rilay.....29

Gambar 4.8. Rangkaian Pengendali Kecepatan Motor.....30

Gambar 4.9. Minimum Sistem Mikrokontroler ATmega8535.....31

Gambar 4.10 Bentuk mekanik pemutar gerabah.....32

Gambar 4.11 Diagram blok kontrol proporsional.....33

Gambar 4.12 Diagram blok kontrol integral.....34

Gambar 4.13 Diagram blok kontrol derivative.....34

Gambar 4.14 diagram blok kontrol PID.....35

Gambar 5.1. Hasil Pengujian Catu Daya Menggunakan Multimeter

untuk *driver motor*.....40

Gambar 5.2 Hasil Pengujian Catu Daya Menggunakan Multimeter

untuk mengaktifkan *coil relay*.....40

Gambar 5.3 Hasil Pengujian Catu Daya Menggunakan Multimeter untuk rangkaian

mikrokontroler.....40

Gambar 5.4. Diagram Blok Pengujian Rangkaian Mikrokontroler.....44

Gambar 5.5. Diagram Alir Program Pengujian Rangkaian

Mikrokontroler.....44

Gambar 5.6. Grafik Respon Pemutar Gerabah tanpa beban.....46

Gambar 5.7. Grafik Respon Pemutar Gerabah dengan Beban 400 gr.....47

Gambar 5.8. Grafik Respon Pemutar Gerabah dengan Beban 600 gr.....47

Gambar 5.9. Grafik Respon Pemutar Gerabah dengan Beban 800 gr.....48

DAFTAR TABEL

Tabel 5.1. Hasil Pengukuran Tegangan Keluaran Rangkaian Sensor <i>Rotary Encoder</i>	41
Tabel 5.2 Hasil Pengujian Driver Motor.....	42
Tabel 5.3 Hasil Pengujian Kecepatan Pemutar Gerabah terhadap Penambahan <i>duty cycle</i> ,.....	43
Tabel 5.4. Hasil Pengujian Rangkaian Mikrokontroler.....	45
Tabel 5.5 Hasil Pengujian Alat Pemutar Gerabah Tanpa Beban.....	46
Tabel 5.6 Hasil Pengujian Alat Pemutar Gerabah Dengan Beban 400gr.....	49
Tabel 5.7 Hasil Pengujian Alat Pemutar Gerabah Dengan Beban 600gr.....	51
Tabel 5.8 Hasil Pengujian Alat Pemutar Gerabah Dengan Beban 800gr.....	54

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kebutuhan dalam mendekorasi interior rumah kini mulai menjadi perhatian khusus, mulai dari interior fungsional hingga interior non fungsional. Gerabah merupakan salah satu contoh karya interior. Gerabah mempunyai bentuk yang bermacam - macam, mulai dari pot, patung, guci hingga pajangan - pajangan kecil. Tak mengherankan jika kemudian gerabah menjadi salah satu produk yang paling digemari untuk mempercantik rumah. Peminat gerabah sangat banyak, hal tersebut terbukti ketika perekenomian indonesia sedang lemah industri gerabah mampu bertahan bahkan dapat berkembang sementara sekian banyak usaha lain mengalami kemacetan.

Terdapat keaneragaman teknik dalam pembuatan gerabah ditiap daerah, tapi pada umumnya teknik yang digunakan adalah dengan membentuk adonan tanah liat pada alat putar yang kemudian dibentuk dengan tangan, teknik ini dapat menghasilkan banyak bentuk yang simetris seperti bulat, silindris, dan bervariasi. Cara pembentukan dengan teknik putar ini sering dipakai oleh para perajin di sentra - sentra gerabah. Perajin gerabah tradisional biasanya menggunakan alat putar tangan (*hand wheel*) atau alat putar kaki (*kick wheel*). Memutar alat putar gerabah secara manual dengan tangan atau dengan kaki memiliki kekurangan yaitu perajin akan kelelahan jika adonan tanah liat yang dibentuk besar.

Berdasarkan permasalahan tersebut maka dalam skripsi ini akan dirancang sebuah sistem elektronika yang mampu memutar alat putar gerabah. Putaran alat putar gerabah ini akan dibuat konstan agar ketika perajin menambah adonan tanah liat maka kecepatan alat putar gerabah tidak melambat. Diharapkan dengan adanya alat ini dapat mempermudah kerja dari para perajin gerabah.

1.2 Rumusan Masalah

Mengacu pada permasalahan yang diuraikan pada latar belakang, maka rumusan masalah dapat ditekankan pada:

1. Bagaimana memanfaatkan pemrograman MIKROKONTROLER ATMEGA8535 dengan menggunakan program CV-AVR sehingga dapat digunakan sebagai pengontrol pada alat ini dengan menggunakan kontroler PID?
2. Bagaimana merancang dan membuat sistem elektronika yang mampu mempertahankan kecepatan putaran pemutar gerabah secara otomatis?

1.3 Batasan Masalah

Mengacu pada permasalahan yang ada maka skripsi ini dibatasi pada:

1. Alat putar gerabah yang dibuat merupakan sebuah model atau *prototype*.
2. Pembahasan ditekankan pada proses mempertahankan kecepatan alat putar gerabah pada kecepatan yang ditentukan. Kinerja *driver* dan rangkaian elektronika tidak dibahas mendalam.
3. Gangguan berupa berat beban material yang berubah-ubah secara acak dalam batasan yang telah ditentukan.
4. Menggunakan sensor *rotary encoder*
5. Menggunakan mikrokontroler ATmega8535
6. Menggunakan kontrol PID sebagai pengaturan.

1.4 Tujuan

Menciptakan sistem elektronika yang dapat mempertahankan kecepatan putaran pemutar gerabah saat terjadi gangguan berupa perubahan berat beban.

1.5 Sistematika Pembahasan

Sistematika penulisan dalam skripsi ini sebaga berikut:

BAB I Pendahuluan

Memuat latar belakang, rumusan masalah, tujuan, batasan masalah dan sistematika pembahasan.

BAB II Teori Penunjang

Membahas teori-teori yang mendukung dalam perencanaan dan pembuatan alat.

BAB III Metodologi

Berisi tentang metode penelitian dan perencanaan alat serta pengujian.

BAB IV Perencanaan dan Pembuatan Alat

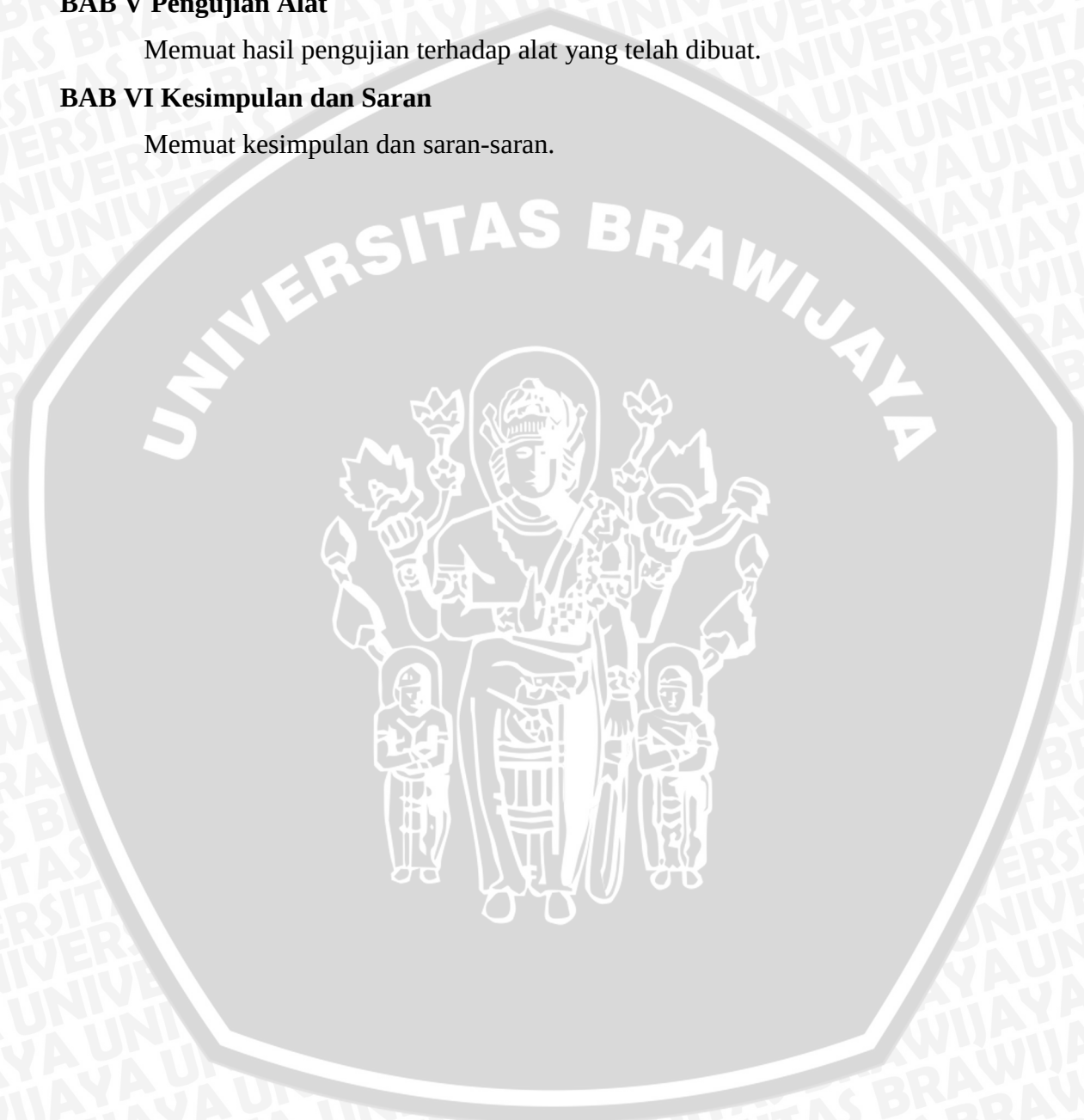
Perancangan alat yang meliputi spesifikasi, perencanaan blok diagram, prinsip kerja dan pembuatan alat.

BAB V Pengujian Alat

Memuat hasil pengujian terhadap alat yang telah dibuat.

BAB VI Kesimpulan dan Saran

Memuat kesimpulan dan saran-saran.



BAB II

TINJAUAN PUSTAKA

Dalam merencanakan dan merealisasikan sistem alat pengatur kecepatan pemutar gerabah dengan kontroler PID menggunakan mikrokontroler ATmega8535 dibutuhkan pemahaman mengenai teori - teori penunjang yang mendukung. Pemahaman ini akan bermanfaat dalam perencanaan dan perealisasi perangkat keras dan perangkat lunak sistem. Teori - teori penunjang tersebut meliputi sensor *rotary encoder* dan mikrokontroler ATmega8535.

2.1 Studi literatur gerabah

Gerabah adalah bagian dari keramik yang dilihat berdasarkan tingkat kualitas bahannya. Namun ada masyarakat yang mengartikan terpisah antara gerabah dan keramik. Ada pendapat gerabah bukan termasuk keramik, karena benda - benda keramik adalah benda-benda pecah belah yang permukaannya halus dan mengkilap seperti porselin dalam wujud vas bunga, guci, dan lain - lain. Sedangkan gerabah adalah barang-barang dari tanah liat dalam wujud seperti periuk, belanga, tempat air, dan lain - lain.

Menurut The Concise Colombia Encyclopedia, kata 'keramik' berasal dari Bahasa Yunani (*Greek*) '*keramikos*' menunjuk pada pengertian gerabah; '*keramos*' menunjuk pada pengertian tanah liat. '*Keramikos*' terbuat dari mineral non metal, yaitu tanah liat yang dibentuk, kemudian secara permanen menjadi keras setelah melalui proses pembakaran pada suhu tinggi. Sedangkan menurut Malcolm G. McLaren dalam Encyclopedia Americana 1996 disebutkan keramik adalah suatu istilah yang sejak semula diterapkan pada karya yang terbuat dari tanah liat alami dan telah melalui perlakuan pemanasan pada suhu tinggi.

Teori tentang keramik lainnya yang terkenal adalah 'teori keranjang'. Teori ini menyebutkan pada zaman prasejarah, keranjang anyaman digunakan orang untuk menyimpan bahan makanan. Agar tidak bocor keranjang tersebut dilapisi dengan tanah liat dibagian dalamnya. Setelah tak terpakai keranjang dibuang keperapian. Kemudian keranjang itu musnah tetapi tanah liatnya yang berbentuk wadah itu ternyata menjadi keras. Teori ini dihubungkan dengan

ditemukannya keramik prasejarah, bentuk dan motif hiasnya di bagian luar berupa relief cap tangan keranjang (Nelson, 1984 : 20).

Dari teori keranjang dan teori lainnya dapat dimengerti bahwa benda-benda keras dari tanah liat dari awal ditemukan sudah dinamakan benda keramik, walaupun sifatnya masih sangat sederhana seperti halnya gerabah dewasa ini. Pengertian ini menunjukkan bahwa gerabah adalah salah satu bagian dari benda - benda keramik.

2.1.1 Sekilas Tentang Proses Pembuatan Gerabah

Proses pembuatan gerabah yang dilakukan perajin gerabah di Indonesia pada umumnya masih dilakukan secara tradisional, berikut merupakan proses pembuatan gerabah:

1. Pengambilan tanah liat. Tanah liat diambil dengan cara menggali secara langsung ke dalam tanah yang mengandung banyak tanah liat yang baik. Tanah liat yang baik berwarna merah coklat atau putih kecoklatan. Tanah liat yang telah digali kemudian dikumpulkan pada suatu tempat untuk proses selanjutnya.
2. Persiapan tanah liat. Tanah liat yang telah terkumpul disiram air hingga basah merata kemudian didiamkan selama satu hingga dua hari. Setelah itu, kemudian tanah liat digiling agar lebih rekat dan liat. Ada dua cara penggilingan yaitu secara manual dan mekanis. Penggilingan manual dilakukan dengan cara menginjak - injak tanah liat hingga menjadi halus. Sedangkan secara mekanis dengan menggunakan mesin giling. Hasil terbaik akan dihasilkan dengan menggunakan proses giling manual.
3. Proses pembentukan. Setelah melewati proses penggilingan, maka tanah liat siap dibentuk sesuai dengan keinginan. Aneka bentuk dan disain dapat dihasilkan dari tanah liat. Seberapa banyak tanah liat dan berapa lama waktu yang diperlukan tergantung pada seberapa besar gerabah yang akan dihasilkan. Perajin gerabah akan menggunakan kedua tangan untuk membentuk tanah liat dan kedua kaki untuk memutar alat pemutar (perbot). Kesamaan gerak dan konsentrasi sangat diperlukan untuk dapat melakukannya. Alat-alat yang digunakan yaitu alat pemutar (perbot), alat

pemukul, batu bulat, kain kecil. Air juga sangat diperlukan untuk membentuk gerabah dengan baik.

4. Penjemuran. Setelah bentuk akhir telah terbentuk, maka diteruskan dengan penjemuran. Sebelum dijemur di bawah terik matahari, gerabah yang sudah agak mengeras dihaluskan dengan air dan kain kecil lalu dibatik dengan batu api. Setelah itu baru dijemur hingga benar-benar kering. Lamanya waktu penjemuran disesuaikan dengan cuaca dan panas matahari.
5. Pembakaran. Setelah gerabah menjadi keras dan benar-benar kering, kemudian banyak gerabah dikumpulkan dalam suatu tempat atau tungku pembakaran. Gerabah - gerabah tersebut kemudian dibakar selama beberapa jam hingga benar-benar keras. Proses ini dilakukan agar gerabah benar - benar keras dan tidak mudah pecah. Bahan bakar yang digunakan untuk proses pembakaran adalah jerami kering, daun kelapa kering ataupun kayu bakar.
6. Penyempurnaan. Dalam proses penyempurnaan, gerabah jadi dapat dicat dengan cat khusus sehingga terlihat indah dan menarik sehingga bernilai jual tinggi.

2.2 Studi Teknik Pengontrolan

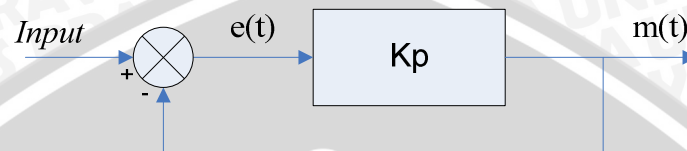
Dalam sistem alat pengatur kecepatan pemutar gerabah dengan kontroler PID menggunakan mikrokontroler ATmega8535 ini masukan berupa kecepatan pemutar gerabah (dalam rpm) dan aktuator berupa motor DC yang menggerakkan pemutar gerabah. Kontroler akan mengatur aktuator agar kecepatan pemutar gerabah sesuai dengan *set point*.

2.2.1 Kontroler PID (*Proportional Integral Derivative*)

2.2.1.1 Kontroler Proporsional

Kontroler proporsional memiliki keluaran yang sebanding/proporsional dengan besarnya sinyal kesalahan (selisih antara besaran yang diinginkan dengan harga aktualnya). Secara lebih sederhana dapat dikatakan, bahwa keluaran kontroler proporsional merupakan perkalian antara konstanta proporsional dengan masukannya. Perubahan pada sinyal masukan akan segera menyebabkan sistem secara langsung mengubah keluarannya sebesar konstanta pengalinya.

Pada Gambar 2.1 menunjukkan blok diagram yang menggambarkan hubungan antara *input* (besaran referensi yang diinginkan), besaran aktual dengan besaran keluaran kontroler proporsional, dan besaran kesalahan (*error*). Sinyal kesalahan (*error*) merupakan selisih antara besaran *setting* dengan besaran aktualnya.



Gambar 2.1 Diagram blok kontroler proporsional

Sumber: Ogata, 1995: 158

Pada pengendali proporsional hubungan antara keluaran kontroler $m(t)$ dan sinyal kesalahan $e(t)$ adalah

$$m(t) = K_p e(t) \quad (2.1)$$

Dengan K_p adalah penguatan proporsional. Keluaran $m(t)$ hanya tergantung pada K_p dan *error*, semakin besar *error* maka semakin besar koreksi yang dilakukan. Penambahan K_p akan menaikkan penguatan sistem sehingga dapat digunakan untuk memperbesar kecepatan respon dan mengurangi kesalahan keadaan mantap.

2.2.1.2 Kontroler Integral

Kontroler integral berfungsi mengurangi kesalahan keadaan mantap yang dihasilkan pada kontroler proporsional sebelumnya. Kalau sebuah plant tidak memiliki unsur integrator ($1/s$), kontroler proporsional tidak akan mampu menjamin keluaran sistem dengan kesalahan keadaan mantapnya nol.

Kontroler integral memiliki karakteristik seperti halnya sebuah integral. Keluaran kontroler sangat dipengaruhi oleh perubahan yang sebanding dengan nilai sinyal kesalahan. Keluaran kontroler ini merupakan jumlahan yang terus menerus dari perubahan masukannya. Kalau sinyal kesalahan tidak mengalami

perubahan, keluaran akan menjaga keadaan seperti sebelum terjadinya perubahan masukan. Gambar 2.2 menunjukkan blok diagram kontroler integral.



Gambar 2.2 Blok diagram kontroler integral

Sumber: Ogata, 1995: 158

Nilai keluaran kontroler $m(t)$ sebanding dengan integral sinyal kesalahan $e(t)$, Sehingga

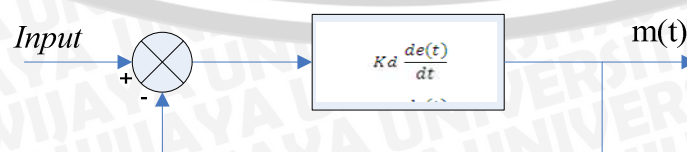
$$\frac{dm(t)}{dt} = Ki \cdot e(t) \quad (2.2)$$

$$m(i) = Ki \int_0^t e(t) dt \quad (2.3)$$

dengan Ki adalah konstanta integral. Jika sinyal kesalahan $e(t)=0$, maka laju perubahan sinyal kendali integral $\frac{dm(t)}{dt} = 0$ atau sinyal keluaran kendali akan tetap berada pada nilai yang dicapai sebelumnya. Aksi control integral digunakan untuk menghilangkan kesalahan posisi dalam keadaan mantap (*error steady state*) tanpa memperhitungkan kecepatan respon.

2.2.1.3 Kontroler Differensial

Kontroler differensial memiliki sifat seperti halnya suatu operasi derivatif. Perubahan yang mendadak pada masukan kontroler, akan mengakibatkan perubahan yang sangat besar dan cepat. Gambar 2.3 menunjukkan blok diagram pada kontroler differensial.



Gambar 2.3 Blok diagram kontroler differensial

Sumber: Ogata, 1995: 158

Nilai keluaran kontroler $m(t)$ sebanding laju sinyal kesalahan $\frac{de(t)}{dt}$.

Hubungan ini dapat ditulis sebagai:

$$m(t) = Kd \frac{de(t)}{dt} \quad (2.4)$$

Kontroler diferensial akan memberikan sinyal kendali keluaran $m(t) = 0$, untuk sinyal kesalahan $e(t)$ yang konstan sehingga kontroler diferensial tidak mempengaruhi keadaan mantap. Kontroler diferensial digunakan untuk memperbaiki atau mempercepat respon transien sebuah sistem serta dapat meredam osilasi.

Berdasarkan karakteristik kontroler tersebut, kontroler diferensial umumnya dipakai untuk mempercepat respon awal suatu sistem, tetapi tidak memperkecil kesalahan pada keadaan tunaknya. Kerja kontroler diferensial hanyalah efek dari lingkup yang sempit, yaitu pada periode peralihan. Oleh sebab itu kontroler differensial tidak bisa digunakan tanpa ada kontroler lain.

2.2.1.4 Kontroler Proporsional Integral (PI)

Aksi kontrolnya dinyatakan dalam persamaan:

$$m(t) = Kp \cdot e(t) + \frac{Kp}{Ti} \int_0^t e(t) dt \quad (2.5)$$

kontroler ini menghasilkan sinyal kesalahan $\int e(t) dt$ kemudian ditambahkan dengan sinyal kesalahan $e(t)$.

2.2.1.5 Kontroler Proporsional Diferensial (PD)

Aksi kontrolnya dinyatakan dalam persamaan:

$$m(t) = Kp \cdot e(t) + Kp \cdot Td \frac{de(t)}{dt} \quad (2.6)$$

Kontroler PD selalu mengukur kemiringan (slope) sinyal kesalahan $\frac{de(t)}{dt}$ dan memperkirakan akan besar overshoot yang akan terjadi serta memberikan

koreksi sebelum terjadi lewatan sebenarnya sehingga diperoleh maximum overshoot yang kecil.

Jika kesalahan keadaan mantap tidak berubah terhadap waktu maka turunannya terhadap waktu sama dengan nol, sehingga kontroler PD tidak mempunyai pengaruh terhadap kesalahan keadaan mantap, tetapi jika terdapat perubahan kesalahan, kontroler PD akan mengurangi besar kesalahan keadaan mantap. Jadi kontroler PD digunakan untuk memperbaiki suatu sistem pengendalian yang tanggapan peralihannya mempunyai maximum overshoot yang berlebihan tanpa memperhitungkan kecepatan responnya.

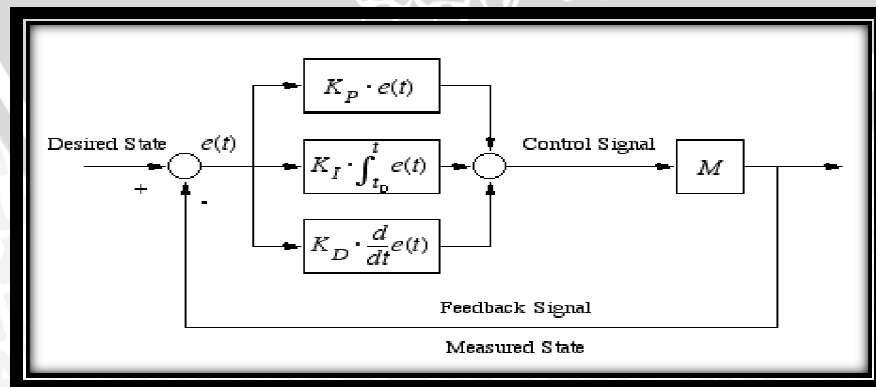
2.2.1.6 Kontroler Proporsional Integral Differensial (PID)

Setiap kekurangan dan kelebihan dari masing-masing kontroler P,I dan D dapat saling menutupi dengan menggabungkan ketiganya secara paralel menjadi kontroler proporsional integral differensial (PID). Elemen-elemen kontroler P,I dan D masing-masing secara keseluruhan bertujuan untuk mempercepat reaksi sebuah sistem, menghilangkan offset dan menghasilkan perubahan awal yang besar (Gunterus, 1994, 8-10). Kontroler PID memiliki diagram kendali seperti yang ditunjukkan dalam Gambar 2.4.

Aksi kontrolnya dinyatakan sebagai:

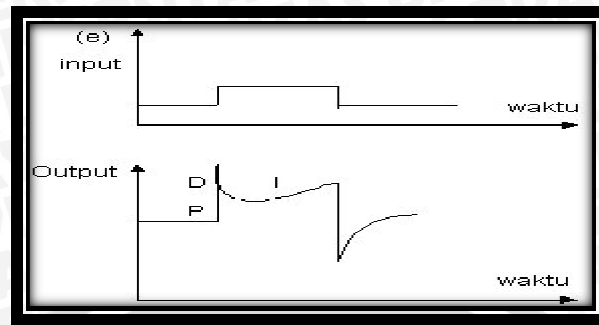
$$m(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt + K_p T_d \frac{de(t)}{dt} \quad (2.7)$$

Jenis kontroler ini digunakan untuk memperbaiki kecepatan respon, mencegah terjadinya kesalahan keadaan mantap serta mempertahankan kestabilan.



Gambar 2.4 Diagram blok kontroler PID

Sumber: Gunterus, 1994:8-11



Gambar 2.5 Hubungan fungsi waktu antara sinyal keluaran dan sinyal masukan kontroler PID

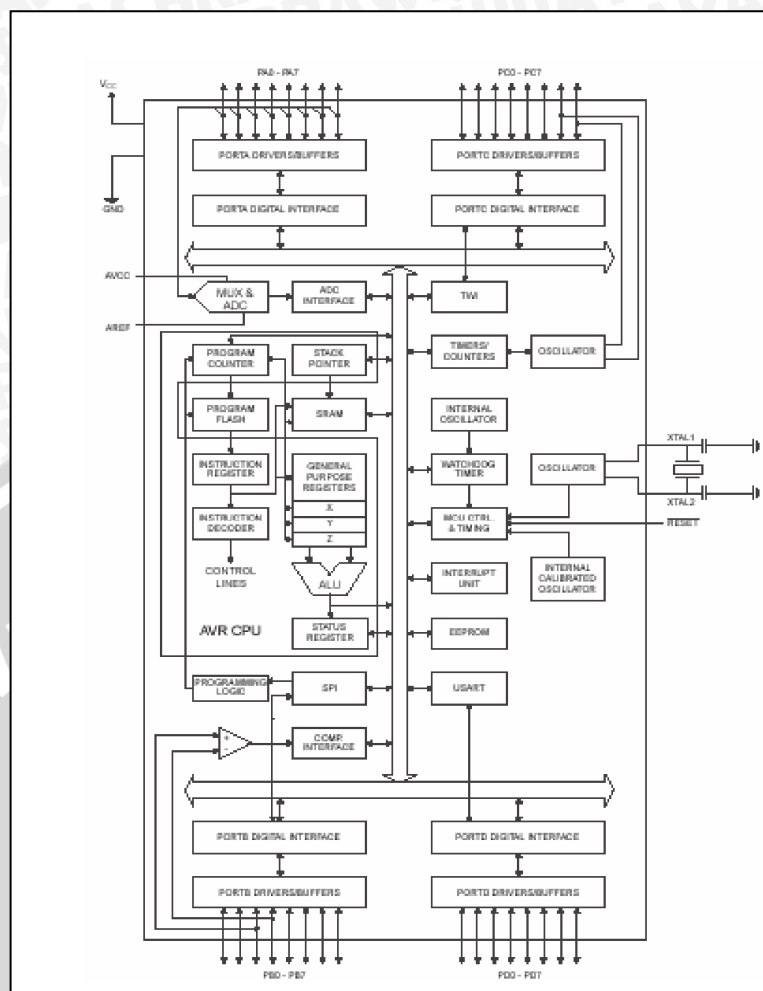
Sumber: Gunterus, 1994:8-11

Keluaran kontroler PID merupakan penjumlahan dari keluaran kontroler proporsional, integral dan differensial. Gambar 2.5 menunjukkan hubungan tersebut. Karakteristik kontroler PID sangat dipengaruhi oleh kontribusi besar dari ketiga parameter P,I dan D. Penyetelan konstanta K_p , T_i dan T_d akan mengakibatkan penonjolan sifat dari masing-masing elemen. Satu atau dua dari ketiga konstanta tersebut dapat disetel lebih menonjol dibanding yang lain. Konstanta yang menonjol itulah yang akan memberikan konstribusi pengaruh pada respon sistem secara keseluruhan (Gunterus, 1994, 8-10).

2.2.2 Mikrokontroler ATmega8535

Mikrokontroler ATmega8535 merupakan Mikrokontroler generasi AVR (*Alf and Vegards Risk processor*). Mikrokontroler AVR memiliki arsitektur RISC (*Reduced Instruction Set Computing*) 8 bit, dimana semua instruksi dikemas dalam kode 16-bit (16-bits word) dan sebagian besar instruksi dieksekusi dalam 1 siklus *clock*.

2.2.2.1 Arsitektur ATmega8535



Gambar 2.6 Diagram Blok Fungsional ATmega8535

Sumber : Data sheet ATmega8535

Gambar 2.6 memperlihatkan bahwa ATmega8535 memiliki bagian sebagai berikut

1. Saluran I/O sebanyak 32 buah, yaitu *Port A*, *Port B*, *Port C*, dan *Port D*.
2. ADC 10 bit sebanyak 8 saluran.
3. Tiga buah *Timer/Counter* dengan kemampuan perbandingan.
4. CPU yang terdiri atas 32 buah *register*.
5. *Watchdog Timer* dengan osilator internal.
6. SRAM sebesar 512 byte.
7. *Memori Flash* sebesar 8 Kb dengan kemampuan *Read While Write*.
8. Unit interupsi internal dan eksternal.
9. Port antarmuka SPI.
10. EEPROM (*Electrically Erasable Programmable Read Only Memory*) sebesar 512 byte yang dapat diprogram saat operasi.

11. Antarmuka komparator analog.
12. Port USART untuk komunikasi serial dengan kecepatan maksimal 2,5 Mbps.
13. Sistem mikroprosesor 8 bit berbasis RISC dengan kecepatan maksimal 16 MHz.

2.2.2.2 Konfigurasi Pin ATmega8535

PDIP			
(XCK/T0) PB0	1	40	PA0 (ADC0)
(T1) PB1	2	39	PA1 (ADC1)
(INT2/AIN0) PB2	3	38	PA2 (ADC2)
(OC0/AIN1) PB3	4	37	PA3 (ADC3)
(SS) PB4	5	36	PA4 (ADC4)
(MOSI) PB5	6	35	PA5 (ADC5)
(MISO) PB6	7	34	PA6 (ADC6)
(SCK) PB7	8	33	PA7 (ADC7)
RESET	9	32	AREF
VCC	10	31	GND
GND	11	30	AVCC
XTAL2	12	29	PC7 (TOSC2)
XTAL1	13	28	PC6 (TOSC1)
(RXD) PD0	14	27	PC5
(TXD) PD1	15	26	PC4
(INT0) PD2	16	25	PC3
(INT1) PD3	17	24	PC2
(OC1B) PD4	18	23	PC1 (SDA)
(OC1A) PD5	19	22	PC0 (SCL)
(ICP1) PD6	20	21	PD7 (OC2)

Gambar 2.7 Konfigurasi Pin ATmega8535

Sumber : Datasheet ATmega8535

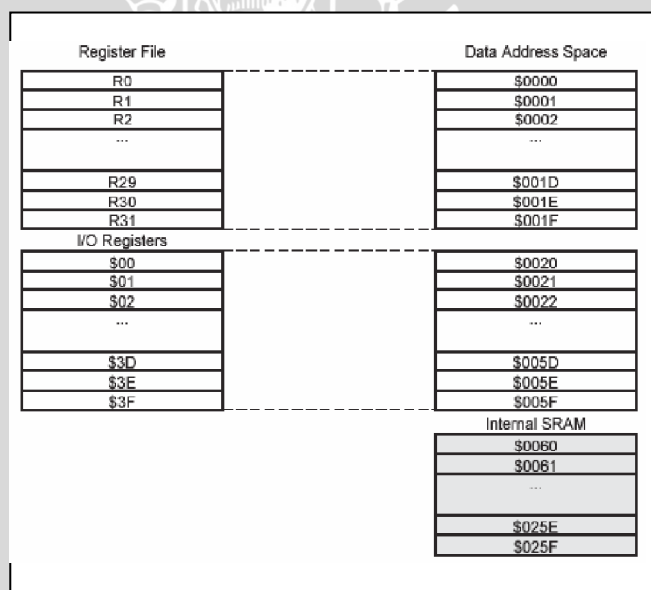
Konfigurasi pin ATmega8535 dapat dilihat dalam Gambar 2.7. Secara fungsional konfigurasi pin ATmega8535 sebagai berikut :

1. VCC merupakan *pin* yang berfungsi sebagai *pin* masukan catu daya.
2. GND merupakan *pin ground*.
3. Port A (PA0..PA7) merupakan *pin* I/O dua arah dan *pin* masukan ADC.
4. Port B (PB0..PB7) merupakan *pin* I/O dua arah dan *pin* fungsi khusus untuk Timer/Counter, Komparator analog, dan SPI.
5. Port C (PC0..PC7) merupakan *pin* I/O dua arah dan *pin* khusus untuk TWI, Komparator analog, dan Timer Oscillator.
6. Port D (PD0..PD7) merupakan *pin* I/O dua arah dan *pin* khusus untuk Komparator analog, Interupsi eksternal, dan Komunikasi serial.

7. *RESET* merupakan *pin* yang digunakan untuk me-reset Mikrokontroler.
8. XTAL1 dan XTAL2 merupakan *pin* masukan *clock* eksternal.
9. AVCC merupakan *pin* masukan tegangan untuk ADC.
10. AREF merupakan *pin* masukan tegangan referensi ADC.

2.2.2.3 Peta Memori

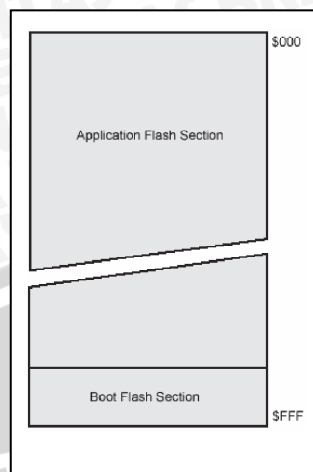
ATmega8535 memiliki ruang pengalamatan memori data dan memori program yang terpisah. Memori data terbagi menjadi 3 bagian, yaitu 32 buah *register* umum, 64 buah *register* I/O, dan 512 *byte* SRAM Internal. Register dengan fungsi umum menempati *space* data pada alamat terbawah, yaitu \$00 sampai \$1F, *register* khusus untuk menangani I/O dan kontrol Mikrokontroler menempati 64 alamat \$20 hingga \$5F, sedangkan SRAM 512 *byte* pada alamat \$60 sampai dengan \$25F. Konfigurasi memori data ditunjukkan Gambar 2.8.



Gambar 2.8. Konfigurasi Memori Data ATmega8535

Sumber : Data sheet ATmega8535

Memori program yang terletak dalam *Flash* PEROM tersusun dalam *word* karena setiap instruksi memiliki lebar 16-bit atau 32-bit. AVR ATmega8535 memiliki 4 Kbyte x 16-bit *Flash* PEROM dengan alamat mulai dari \$000 sampai \$FFF seperti ditunjukkan pada Gambar 2.9. AVR memiliki 12-bit *Program Counter* (PC) sehingga mampu mengalami isi *Flash*.



Gambar 2.9. Memori Program ATmega8535

Sumber : Data sheet ATmega8535

ATmega8535 juga memiliki memori data berupa EEPROM 8-bit sebanyak 512 *byte*. Alamat EEPROM dimulai dari \$000 sampai \$1FF. 2.3.4. Status Register (SREG). Status Register merupakan register berisi status yang dihasilkan pada setiap operasi yang dilakukan ketika suatu instruksi dieksekusi. Status Register ditunjukkan pada Gambar 2.10. SREG merupakan bagian dari inti CPU Mikrokontroler.

Bit	7	6	5	4	3	2	1	0	
	I	T	H	S	V	N	Z	C	SREG
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Gambar 2.10. Status Register ATmega8535

Sumber : Data sheet ATmega8535

a. Bit 7 – I : *Global Interrupt Enable*

Bit yang harus di *set* untuk meng-*enable* interupsi.

b. Bit 6 – T : *Bit Copy Storage*

Instruksi BLD dan BST menggunakan bit-T sebagai sumber atau tujuan dalam operasi *bit*. Suatu *bit* dalam sebuah *register* GPR dapat disalin ke *bit* T menggunakan instruksi BST, dan sebaliknya bit-T dapat disalin kembali ke suatu bit dalam register GPR menggunakan instruksi BLD.

c. Bit 5 – H : *Half Carry Flag*

d. Bit 4 – S : *Sign Bit*

Bit-S merupakan hasil operasi EOR antara *flag-N (negative)* dan *flag-V (two's complement overflow)*.

e. Bit 3 – V : *Two's Complement Overflow Flag*

Bit yang berguna untuk mendukung operasi aritmatika.

f. Bit 2 – N : *Negative Flag*

Bit akan diset bila suatu operasi menghasilkan bilangan negatif.

g. Bit 1 – Z : *Zero Flag*

Bit akan diset bila hasil operasi yang diperoleh adalah nol.

h. Bit 0 – C : *Carry Flag*

Bit akan diset bila suatu operasi menghasilkan *carry*.

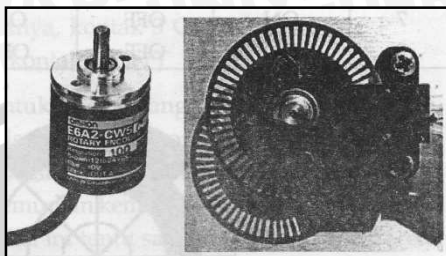
2.3 Studi Sensor

Sensor rotary encoder

Sensor putaran (*rotary encoder*) digunakan untuk mendeteksi banyak putaran pada motor DC. Putaran motor pada aplikasi ini dihitung setiap selang waktu tertentu. Dengan mengetahui putaran motor penggerak robot maka dapat ditentukan pergerakan, kecepatan, percepatan atau sudut putaran. Salah satu sensor *rotary* yang sering digunakan adalah jenis optik. Sensor ini terdiri dari *Light Emitting Diode (LED)* inframerah dan *phototransistor*. Secara prinsip sensor mendeteksi halangan diantara LED inframerah dan *phototransistor*. Apabila tidak terhalang maka cahaya dari LED akan mengenai transistor dan transistor akan aktif. Apabila cahaya LED terhalang maka transistor tidak aktif.

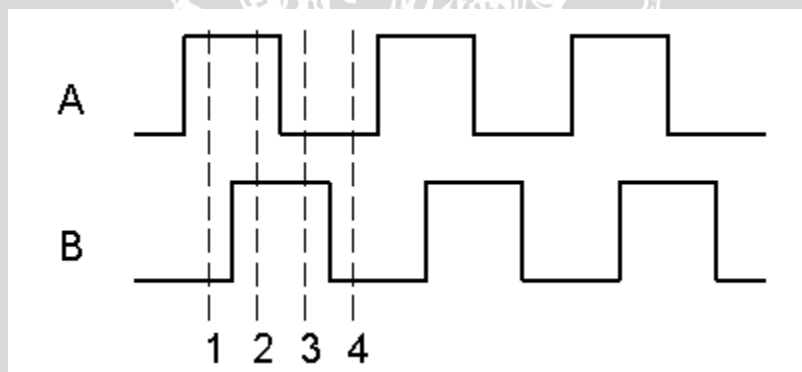
Penghalang cahaya pada sensor optocoupler ini dapat terjadi dengan menggunakan sebuah piringan yang terhubung dengan poros motor. Pada piringan tersebut perlu diberikan lubang-lubang yang akan dilewati sinar inframerah pada posisi-posisi tertentu. Untuk menambah akurasi perhitungan dapat dilakukan

dengan menambah jumlah lubang yang ada pada piringan atau dengan menggunakan roda gigi percepatan. Adapun ilustrasi sensor rotari seperti ditunjukkan dalam Gambar 2.11.



Sumber: Riyanto Sigit, 2007:44.

Dengan menggunakan dua buah *optocoupler* dengan posisi bergeser 90° maka didapatkan sinyal keluaran dari sensor berupa sinyal *quadrature*. Sinyal *quadrature* terdiri dari dua sinyal periodik dengan beda fasa sebesar 90° sehingga dapat dideteksi arah putaran pada piringan berlubang. Ilustrasi sinyal *quadrature* ditunjukkan dalam Gambar 2.12.



Gambar 2.12 Ilustrasi Sinyal Quadrature

Sumber: Borenstein, 1996: 14.

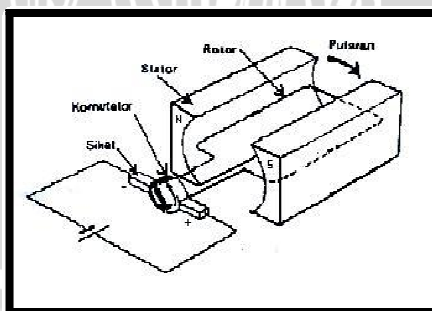
2.4 Motor DC

DC motor yang digunakan dalam penelitian adalah DC motor yang menggunakan permanen magnet. Alasan pemilihan DC motor tipe ini adalah kemudahan dalam pengontrolan dengan menggunakan pengaturan tegangan DC. Medan stator motor jenis ini dihasilkan oleh magnet permanen bukan

elektromagnet. DC motor mempunyai kurva kecepatan torsi yang linier dalam jangka yang lebar. Penggunaan magnet permanen tidak membutuhkan daya listrik untuk menghasilkan medan stator, sehingga daya dan pendinginan yang diperlukan lebih rendah dibandingkan motor yang menggunakan elektromagnet. Perubahan kecepatan motor dapat diatur dengan cara mengubah-ubah besarnya tegangan DC yang diberikan. DC motor memiliki beberapa keunggulan, yaitu:

- Bentuknya kompak, ringan dan berdaya kerja tinggi
- Dapat bekerja pada daerah atau tempat yang mempunyai lingkungan ekstrem
- Biaya perawatan mudah

Motor DC hampir sama konstruksinya dengan motor AC, perbedaannya terletak pada sikat dan cincin belah (komutator). Saat siklus pertama, arus mengalir dari kutub positif ke negatif. Aliran arus yang melewati bagian kabel yang berada didekat kutub N magnet akan menimbulkan gaya Lorentz ke bawah. Sementara itu aliran arus yang melewati kabel yang berada di dekat kutub S magnet akan menyebabkan gaya Lorentz ke atas. Kedua perpaduan gaya Lorentz tersebut akan menyebabkan kawat berputar. Pada siklus berikutnya terjadi hal yang serupa seperti pada siklus sebelumnya. Apabila arus terus-menerus dialirkan, maka kawat akan berputar secara terus menerus pula. Pada aplikasi sesungguhnya, kawat adalah sebuah rotor yang akan dikopel dengan sebuah as dan akan memutar as tersebut terus menerus seiring perputaran motor. Motor DC ditunjukkan dalam Gambar 2.13.



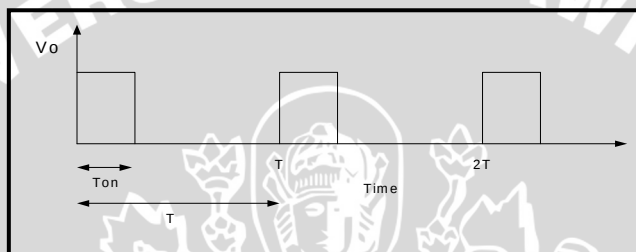
Gambar 2.13 Motor DC

Sumber: Kismet F [1994:98]

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. Pada AVR ATmega8535 *duty cycle* ditentukan dari *Output Compare Register 0* (OCR0). Sinyal PWM (*Pulse width Modulation*) secara umum dapat dilihat dalam Gambar 2.14.



Gambar 2.14 Sinyal PWM Secara Umum

Sumber: Microchip, 1995

$$\text{Duty cycle} = \frac{T_{on}}{T} \times 100\% \quad (2.8)$$

$$V_{DC} = \text{Duty cycle} \times V_{CC} \quad (2.9)$$

Sedangkan frekuensi sinyal dapat ditentukan dengan rumus sebagai berikut:

$$f_{OCn} = \frac{f_{clk_I/O}}{N \cdot 256} \quad (2.10)$$

Timer/counter yang digunakan pada PWM ini yaitu *Timer/Counter0* (8-bit) dengan mode Fast PWM dan *Prescalers factor* (N) yaitu 256.

BAB III

METODOLOGI PENELITIAN

Dalam perealisasiian tujuan skripsi dibutuhkan metodologi dalam pelaksanaannya. Berikut ini adalah metodologi yang digunakan dalam penelitian:

3.1 Studi Literatur

Literatur yang dibutuhkan adalah dasar teori yang berhubungan dengan alat yang akan dirancang, yaitu sebagai berikut:

1. Motor DC
2. Mikrokontroler
3. Sensor *Rotary Encoder*

3.2 Perancangan Alat

Pada tahap perancangan alat, dibuat suatu blok diagram fungsional dari rangkaian yang direncanakan. Perancangan mekanik dilakukan sesuai desain untuk mempermudah pembuatan. Perancangan rangkaian dilakukan pada tiap-tiap blok untuk mempermudah perancangan serta penentuan nilai komponen yang digunakan. Perancangan perangkat lunak dilakukan dengan membuat diagram alir untuk program utama dan sub program. Secara garis besar perancangan alat dilakukan dalam tahap berikut:

- 1) Penentuan spesifikasi alat.
- 2) Pembuatan diagram blok sistem keseluruhan.
- 3) Perancangan mekanik.
- 4) Perancangan perangkat elektrik yang terdiri atas rangkaian mikrokontroler dan rangkaian sensor.

3.3 Pembuatan Alat

Pembuatan alat meliputi pembuatan perangkat mekanik sebagai pemutar gerabah, perangkat elektrik sebagai komponen utama serta pembuatan perangkat lunak sebagai komponen pendukung pengujian.

3.3.1 Pembuatan pemutar gerabah

Pembuatan pemutar gerabah dilakukan dengan pembuatan desain pemutar gerabah terlebih dahulu kemudian dilanjutkan dengan pembuatan kerangka pemutar gerabah dan penyusunan motor DC pada pemutar gerabah.

3.3.2 Pembuatan Perangkat Keras

Pembuatan alat dengan menggunakan komponen elektronika yang telah direncanakan. Pembuatan alat untuk perangkat keras juga meliputi pembuatan driver motor, pembuatan rangkaian mikrokontroler, pembuatan rangkaian sensor, dan pengkabelan semuanya.

3.3.3 Pembuatan Perangkat Lunak

Untuk pembuatan software dilakukan dengan pembuatan flowchart terlebih dahulu kemudian dibuat programnya.

3.4 Pengujian Alat

Untuk mengetahui apakah alat yang telah dibuat sesuai dengan yang direncanakan maka dilakukan pengujian rangkaian. Pengujian yang dilakukan terbagi dua, yaitu :

1) Pengujian perangkat keras

Perangkat keras yang telah dibuat tahap demi tahap akan diuji satu persatu sesuai blok diagram. Pengujian perangkat keras ini meliputi pengujian kestabilan putaran motor DC pada pemutar gerabah dari berbagai *disturbance* yang ada.

2) Pengujian secara keseluruhan

Pengujian secara keseluruhan dilakukan dengan menghubungkan tiap perangkat keras sesuai dengan blok diagram dan menjalankan perangkat lunaknya.

3.5 Pengambilan Kesimpulan dan Saran

Tahap terakhir adalah pengambilan kesimpulan dan saran. Kesimpulan diperoleh dari hasil pengujian alat dan kesesuaiannya dengan teori yang telah dipelajari. Saran diberikan untuk memperbaiki kesalahan, dan kemungkinan pengembangan alat agar lebih baik untuk penelitian selanjutnya.



BAB IV

PERANCANGAN DAN PEMBUATAN ALAT

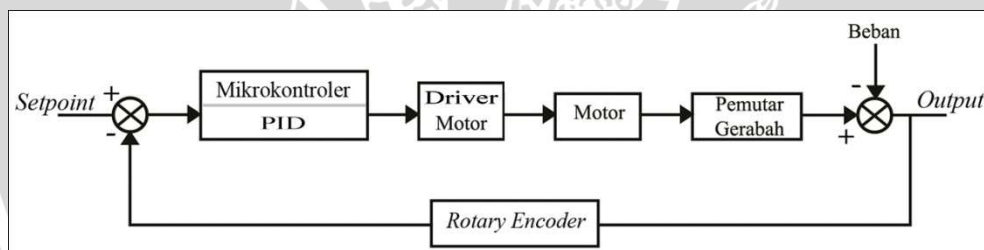
Bab ini menjelaskan mengenai spesifikasi alat, perancangan perangkat keras dari penyetabilan kecepatan pemutar gerabah dengan kontroler PID. Perancangan alat ini dilakukan secara bertahap dalam bentuk blok sehingga akan memudahkan dalam analisis pada setiap bloknnya maupun secara keseluruhan.

Perancangan ini terdiri dari:

- Perancangan blok diagram sistem.
- Perancangan perangkat keras (catu daya sistem, sensor *rotary encoder*, driver motor, dan rangkaian mikrokontroler).
- Perancangan bentuk mekanik pemutar gerabah.
- Perancangan sistem logika *PID* serta implementasinya pada mikrokontroler.

4.1. Perancangan Sistem

Blok diagram sistem yang dirancang ditunjukkan dalam Gambar 4.1.



Gambar 4.1. Blok Diagram Sistem

Pemutar gerabah pada sistem ini digerakkan oleh satu motor DC. Pada poros pemutar gerabah diberi *gear*, dimana *gear* ini terhubung dengan *gear* yang terdapat pada motor DC menggunakan *belt*. Ketika motor DC berputar maka *belt* akan memutar poros dari pemutar gerabah.

Pada poros pemutar gerabah dipasang sebuah sensor *rotary encoder* yang berfungsi menghitung kecepatan putaran pemutar gerabah. Sensor ini terhubung dengan Mikrokontroler ATmega8535 sebagai kontrolernya.

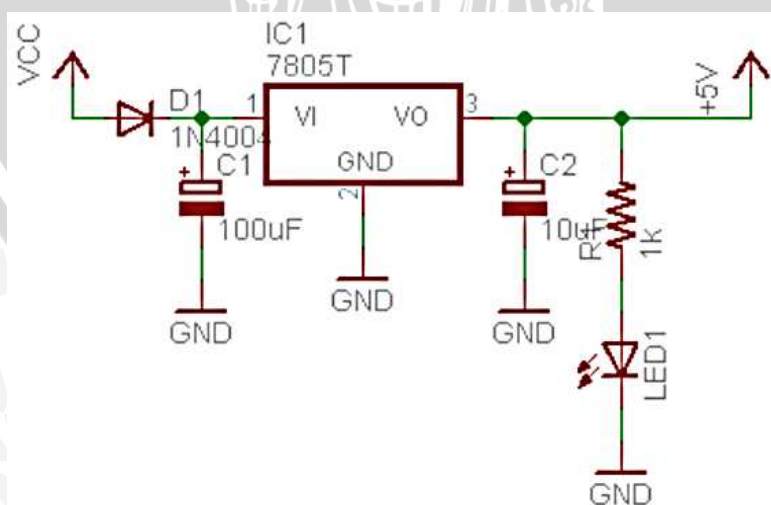
Mikrokontroler ATmega8535 berfungsi sebagai PID kontroler yang membaca dan menghitung kecepatan dari putaran pemutar gerabah. Kemudian data tersebut dibandingkan dengan *set point* untuk kemudian dicari nilai *error* nya dan dilakukan penghitungan untuk nilai kontroler yang diperlukan. Mikrokontroler ini terhubung dengan *driver* motor yang berfungsi sebagai pengkondisi sinyal dari sinyal kontrol mikrokontroler menjadi sinyal untuk menggerakkan aktuatornya yaitu motor DC. *Driver* motor ini berfungsi mengeksekusi hasil perhitungan dari mikrokontroler.

4.2. Perancangan Perangkat Keras

4.2.1. Rangkaian Catu Daya Sistem

Sistem mikrokontroler membutuhkan catu daya sebesar 5 volt agar dapat bekerja. Catu daya 5 volt sudah memenuhi tegangan kerja mikrokontroler ATmega8535 yang berkisar antara 4,5 volt sampai 5,5 volt berdasarkan *datasheet*.

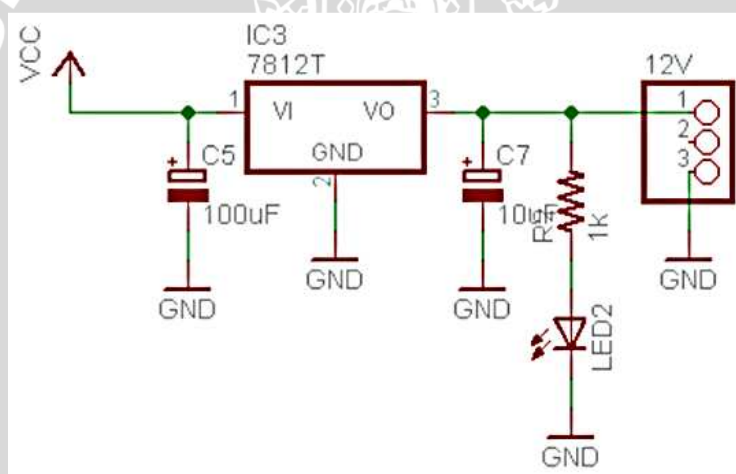
Rangkaian catu daya ini menggunakan *Fixed Output Regulator* sesuai pada *datasheet* LM78XX. Regulator yang digunakan adalah jenis LM7805 yang memiliki tegangan keluaran minimal 4,8 volt dan maksimal 5,2 volt berdasarkan *datasheet* nya. Karena tegangan catu yang dibutuhkan adalah 5 volt dan tegangan keluaran regulator adalah 4.9 volt, maka tegangan untuk mikrokontroler sudah tercukupi. Rangkaian catu daya untuk mikrokontroler ditunjukkan dalam Gambar 4.2.



Gambar 4.2. Rangkaian Catu Daya untuk mikrokontroler

Tegangan sumber yang digunakan dalam rangkaian catu daya adalah 5 volt. Nilai keluaran regulator langsung berhubungan dengan mikrokontroler, rangkaian sensor, dan juga rangkaian *driver*.

Pada rangkaian *driver* digunakan *relay* dengan *coil* yang membutuhkan tegangan sebesar 12 volt. Untuk itu digunakan satu rangkaian lagi agar tegangan 12 volt untuk *coil relay* terpenuhi. Rangkaian ini menggunakan *Fixed Output Regulator* sesuai pada *datasheet* LM78XX. Regulator yang digunakan adalah jenis LM7812 yang memiliki tegangan keluaran minimal 11,5 volt dan maksimal 12,5 volt berdasarkan *datasheet* nya. Tegangan itu sudah cukup untuk mengaktifkan *coil relay* yang membutuhkan tegangan 12 volt. Rangkaian catu daya untuk *coil relay* ditunjukkan dalam Gambar 4.3.

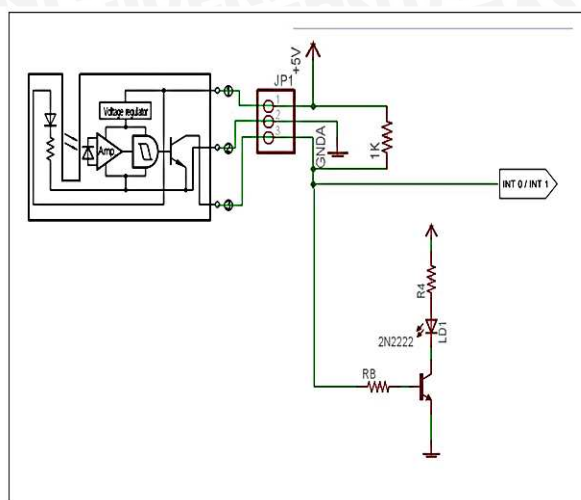


Gambar 4.3. Rangkaian Catu Daya Coil Rilay

4.2.2. Sensor Rotary Encoder

Sensor putaran (*rotary encoder*) yang digunakan adalah sensor putaran berbasis optik menggunakan komponen *Optical IC* dan piringan berpola. Piringan berpola ini diletakkan pada poros pemutar gerabah, sensor *rotary encoder* diletakkan sedemikian rupa agar tak mengganggu pergerakan piringan berpola. Komponen *Optical IC* yang digunakan adalah tipe GP1A25LC produksi SHARP Corporation. Komponen ini merupakan kombinasi LED inframerah sebagai pemancar cahaya, regulator tegangan, penguat, dan *phototransistor* NPN

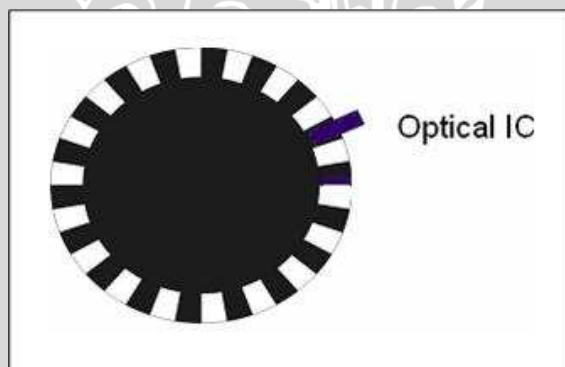
sebagai penerima cahaya. Gambar 4.4 menunjukkan rangkaian sensor *rotary encoder*.



Gambar 4.4. Rangkaian Sensor Rotary Encoder

Resolusi sensor yang digunakan pada perancangan ini sebesar 58 pulsa untuk setiap putaran penuh.

Sifat sinyal *incremental* didapatkan dengan membuat pola berlubang pada sisi luar piringan berpola seperti ditunjukkan dalam Gambar 4.5.

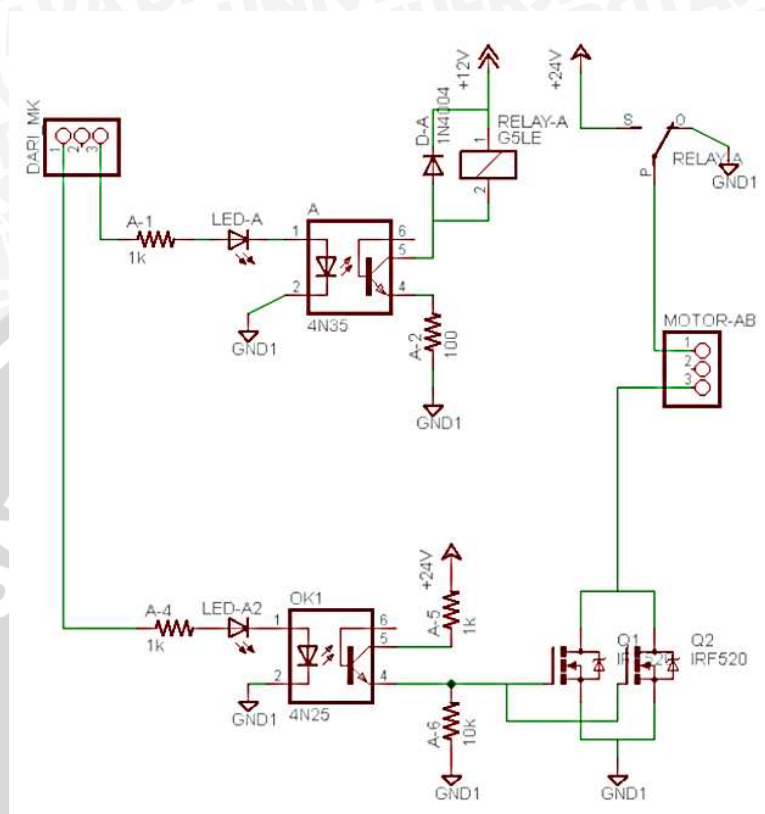


Gambar 4.5. Letak *Optical IC* Pada Piringan Berpola

4.2.3. Rangkaian Driver Motor

Driver pada rangkaian ini menggunakan *relay* untuk logika ON-OFF nya dan Untuk mengontrol putaran motor digunakan E-MOSFET kanal N dengan

masukannya berupa sinyal PWM dari mikrokontroler. Berikut ini rangkaian *driver* motor ditunjukkan dalam Gambar 4.6.



Gambar 4.6 Rangkaian driver motor

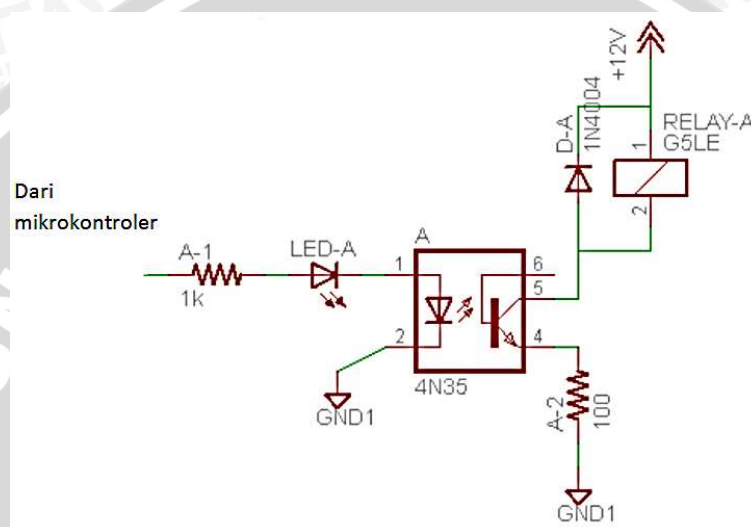
Pada rangkaian *driver* digunakan *optocoupler* 4N25 yang berfungsi untuk memisahkan tegangan 5 volt mikrokontroler dengan tegangan 12 volt untuk *coil relay*. *Optocoupler* ini juga berfungsi sebagai saklar untuk mengaktifkan *coil relay* dengan logika dari mikrokontroler.

Optocoupler 4N25 memiliki spesifikasi sebagai berikut:

- *turn on time* = $2\mu\text{s}$
- *turn off time* = $2\mu\text{s}$
- $I_{F\text{max}} = 60\text{mA}$
- $V_F = 1,2\text{ V}$
- $I_{C\text{max}} = 100\text{mA}$
- $V_{C\text{esat}} = 0,3\text{ V}$

Alasan pemilihan 4N25 sebagai komponen *optocoupler* yang akan digunakan adalah karena 4N25 memiliki *turn on time* dan *turn off time* cukup kecil yaitu 2μs. Dengan *turn on time* dan *turn off time* yang kecil memungkinkan *optocoupler* meneruskan sinyal dengan frekuensi tinggi.

Rangkaian pemisah optocoupler serta perancangan nilai resistor untuk driver NPN ditunjukkan dalam Gambar 4.7.



Gambar 4.7. Rangkaian Pemisah Optocoupler Untuk Coil Rilay

Pengontrol kecepatan menggunakan ATmega8535, dengan spesifikasi:

- $I_{OL} = 20 \text{ mA}$
- $I_{OH} = 20 \text{ mA}$
- $V_{OL} = 0 \text{ V}$
- $V_{OH} = 5 \text{ V}$

Untuk perancangan nilai resistor LED *optocoupler* sebagai berikut:

$$\begin{aligned}
 R_{led} &= \frac{V_A - V_F}{I_{OH}} \\
 &= \frac{5 \text{ V} - 1,2 \text{ V}}{20 \times 10^{-3}} \\
 &= \frac{3,8 \text{ V}}{20 \times 10^{-3}} \\
 &= 190 \Omega
 \end{aligned}
 \tag{4.1}$$

R yang dipakai dalam perancangan adalah 330Ω.

$$\begin{aligned}
 I_{OH} &= \frac{V_A - V_F}{R_{led}} & (4.2) \\
 &= \frac{5\text{ V} - 1,2\text{ V}}{330} \\
 &= \frac{3,8\text{ V}}{330} \\
 &= 11,5\text{ mA}
 \end{aligned}$$

Jadi arus yang mengalir dalam LED *optocoupler* sebesar 11,5 mA. Masih di bawah arus maksimal *pinout* mikrokontroler yang sebesar 20mA.

Perancangan nilai resistor basis transistor darlington NPN TIP 142 sebagai berikut:

Analisis loop tertutup:

$$V_{cc} - V_{ce_{sat}} - V_B - V_{be_{sat}} = 0 \quad (4.3)$$

$$V_B = V_{cc} - V_{ce_{sat}} - V_{be_{sat}}$$

$$R_B = \frac{V_{cc} - V_{ce_{sat}} - V_{be_{sat}}}{I_{B_{max}}}$$

$$= \frac{V_{cc} - V_{ce_{sat}} - V_{be_{sat}}}{I_{C_{max}} / \beta_{min}}$$

$$= \frac{20,7 - 0,3 - 3}{10 / 500}$$

$$= \frac{20,7}{0,002}$$

$$= 1035\text{ ohm}$$

R yang digunakan dalam perancangan adalah 1Kohm.

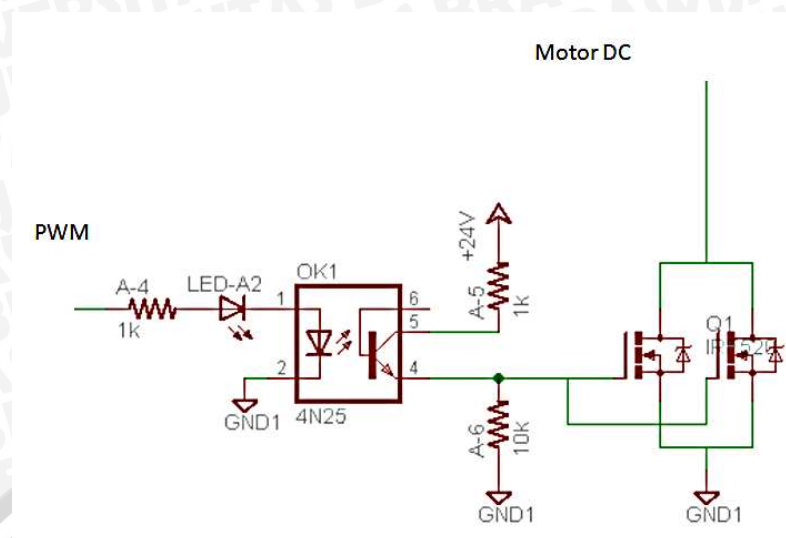
$$I_{B_{max}} = \frac{V_{cc} - V_{ce_{sat}} - V_{be_{sat}}}{1\text{ Kohm}}$$

(4.4)

$$= \frac{20,7\text{ V}}{1\text{ Kohm}}$$

$$= 0,0207\text{ A}$$

Untuk mengontrol putaran motor digunakan E-MOSFET kanal N dengan masukan berupa sinyal PWM dari mikrokontroler. Rangkaian pengendali kecepatan motor ditunjukkan dalam Gambar 4.8.



Gambar 4.8. Rangkaian Pengendali Kecepatan Motor

FET yang digunakan adalah IRFZ44N dengan spesifikasi sebagai berikut:

- $V_{DS\ max} = 55V$
- $I_{D\ max} = 49A$
- $R_{DS\ on\ max} = 22\Omega$
- $V_{GS\ max} = 20V$
- $V_{GS\ threshold} = 3V$

Alasan pemilihan IRFZ44N sebagai komponen FET 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.

Untuk aplikasi pengontrol kecepatan putaran motor, FET selalu dikondisikan dalam keadaan saturasi atau *cut off*-nya. Hal ini dimaksudkan agar tidak terlalu banyak daya yang terbuang dalam FET itu sendiri.

Perancangan nilai resistor R untuk tegangan masukan E-MOSFET:

$$R_{opto} = \frac{24 - V_{CEsat}}{I_C} \quad (4.5)$$

$$= \frac{24 - 0,3}{100 \cdot 10^{-3}} = \underline{\underline{237\Omega}}$$

Nilai R_{opto} yang akan digunakan dalam sistem adalah 270Ω .

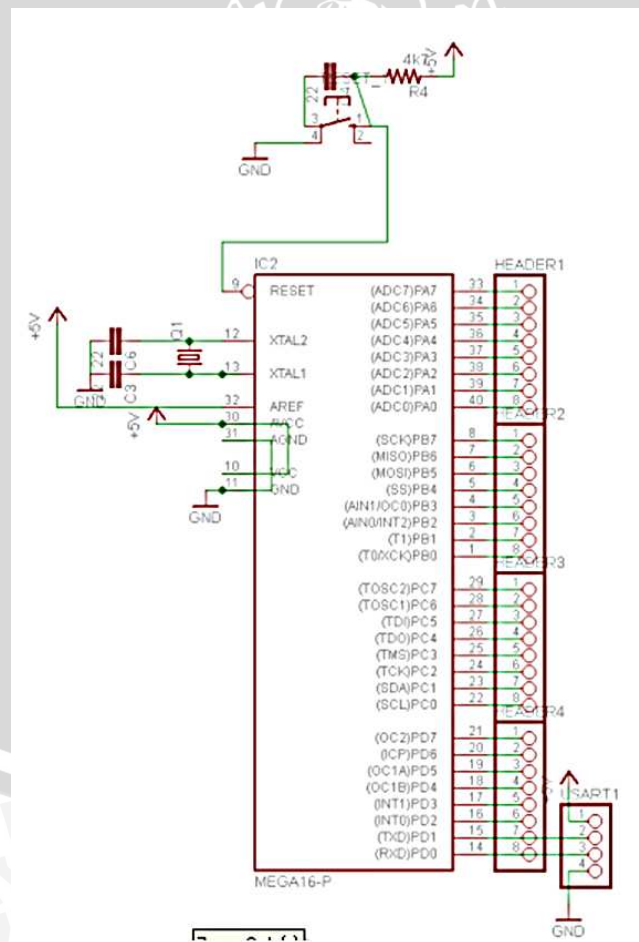
$$I_C = \frac{24 - V_{CEsat}}{R_{8,12}}$$

$$I_C = \frac{24 - 0,3}{270} = 8,78 \cdot 10^{-2} = 87,8mA$$

(4.6)

4.2.4. Rangkaian Mikrokontroler

Pada skripsi ini menggunakan mikrokontroler ATmega8535 sebagai pengolah utama sebagai pengolah data dari sensor rotari, kecepatan motor serta melakukan proses logika *PID*. Konfigurasi kaki I/O dari mikrokontroler ATmega8535 ditunjukkan dalam Gambar 4.9.



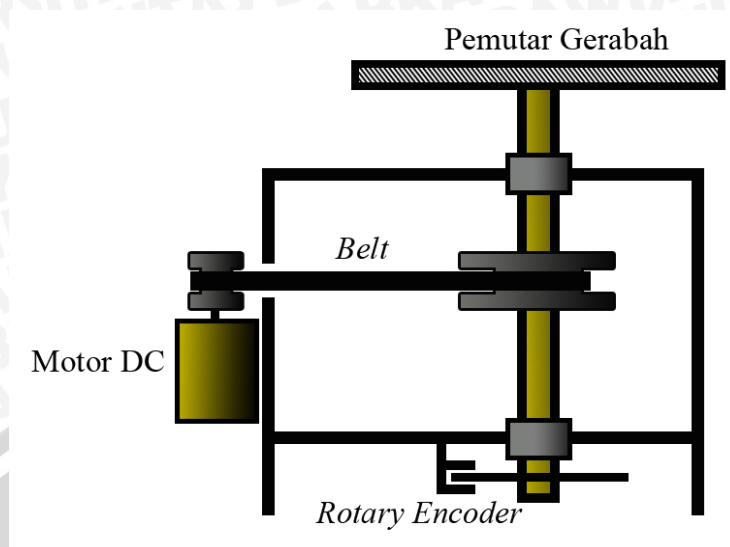
Gambar 4.9. Minimum Sistem Mikrokontroler ATmega8535

Mikrokontroler ATmega8535 memiliki kaki sebanyak 40 pin, 32 diantaranya dapat diprogram menjadi masukan atau keluaran. Berikut ini adalah pin-pin yang digunakan dalam perancangan ini.

PIND.2	= digunakan sebagai masukan sensor putaran (sensor rotari)
PIND.5	= digunakan untuk mengaktifkan <i>driver</i> motor
PIND.6	= digunakan untuk mengatur besar kecepatan motor
XTAL1	= digunakan sebagai masukan dari rangkaian osilator kristal
XTAL2	= digunakan sebagai masukan dari rangkaian osilator kristal

4.3. Perancangan Bentuk Mekanik Pemutar Gerabah

Rangka pemutar gerabah ini terbuat dari kayu berukuran Panjang=43cm, Lebar=25cm, dan Tinggi=30cm. Di bagian atasnya terdapat tempat beban berbentuk lingkaran yang berdiameter 20 cm, pada poros lingkaran ini diletakkan sebuah besi berdiameter 8 mm. Pada besi poros pemutar gerabah dipasang sebuah *gear*, dimana *gear* ini terhubung dengan *gear* yang terdapat pada motor DC dengan menggunakan *belt*. Ketika motor DC berputar maka *belt* akan memutar poros dari pemutar gerabah. Pada poros pemutar gerabah paling bawah dipasang sebuah sensor *rotary encoder* yang berfungsi menghitung kecepatan putaran pemutar gerabah. Sensor ini terhubung dengan Mikrokontroler ATmega8535 sebagai kontrolernya. Bentuk mekanik pemutar gerabah ditunjukkan dalam Gambar 4.10.



Gambar 4.10 Bentuk mekanik pemutar gerabah

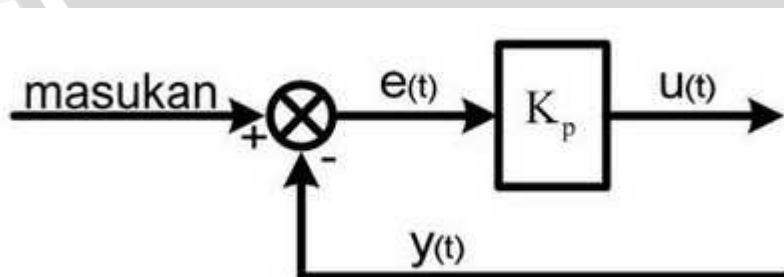
4.4. Perancangan sistem logika *PID* serta implementasinya pada mikrokontroler.

Kontrol Proporsional

Kontrol proporsional berfungsi untuk memperkuat sinyal kesalahan penggerak (sinyal *error*), sehingga akan mempercepat keluaran sistem mencapai titik referensi. Hubungan antara *input* kontroler $u(t)$ dengan sinyal *error* $e(t)$ terlihat pada persamaan 4.7.

$$u(t) = K_p e(t) \quad (4.7)$$

K_p adalah konstanta proporsional. Diagram blok kontrol proporsional ditunjukkan dalam Gambar 4.11.



Gambar 4.11 diagram blok kontrol proporsional

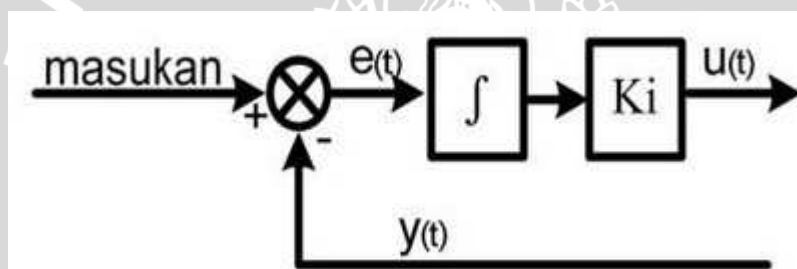
Sumber: Ogata, 1995: 158

Kontrol Integral

Kontrol integral pada prinsipnya bertujuan untuk menghilangkan kesalahan keadaan tunak (*offset*) yang biasanya dihasilkan oleh kontrol proporsional. Hubungan antara *output* kontrol integral $u(t)$ dengan sinyal *error* $e(t)$ terlihat pada persamaan 4.8.

$$u(t) = K_i \int_0^t e(t) dt \quad (4.8)$$

K_i adalah konstanta integral. Diagram blok kontrol integral ditunjukkan dalam Gambar 4.12.



Gambar 4.12 diagram blok kontrol integral

Sumber: Ogata, 1995: 158

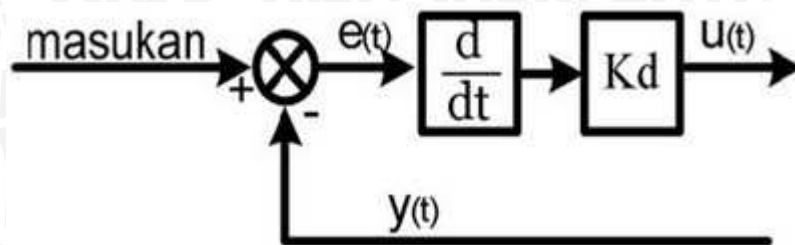
Kontrol Derivatif (turunan)

Kontrol derivatif dapat disebut pengendali laju, karena *output* kontroler sebanding dengan laju perubahan sinyal *error*. Hubungan antara *output* kontrol derivatif $u(t)$ dengan sinyal *error* $e(t)$ terlihat pada persamaan 4.9.

$$u(t) = K_d \frac{de(t)}{dt} \quad (4.9)$$

Blok kontrol derivatif ditunjukkan dalam Gambar 4.13. Kontrol derivatif tidak akan pernah digunakan sendirian, karena kontroler ini hanya akan aktif pada periode peralihan. Pada periode peralihan, kontrol derivatif menyebabkan adanya

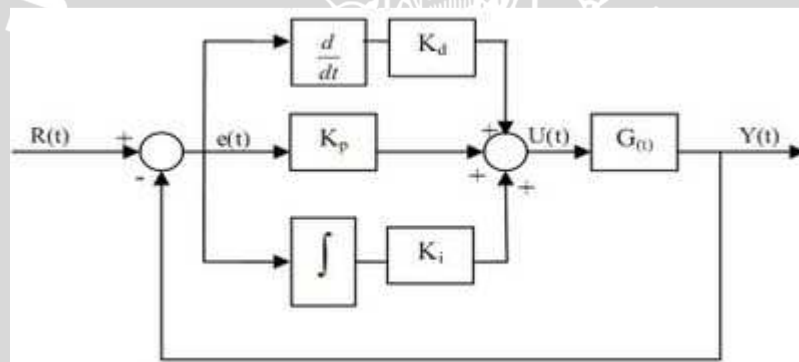
redaman pada sistem sehingga lebih memperkecil lonjakan. Seperti pada kontrol proporsional, kontrol derivatif juga tidak dapat menghilangkan *offset*.



Gambar 4.13 diagram blok kontrol derivatif

Sumber: Ogata, 1995: 158

Gabungan dari ketiga kontroler tersebut menjadi **kontrol PID**. Diagram Blok dari kontrol PID ditunjukkan dalam Gambar 4.14.



Gambar 4.14 diagram blok kontrol PID

Sumber: Ogata, 1995: 158

Sehingga persamaan untuk kontrol PID adalah:

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (4.10)$$

atau

$$u(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right) \quad (4.11)$$

Dengan:

$u(t)$ = sinyal *output* pengendali PID

K_p = konstanta proporsional

T_i = waktu integral

T_d = waktu derivatif

K_i = konstanta integral (K_p/T_i)

K_d = konstanta derivatif ($K_p.T_d$)

$e(t)$ = sinyal *error* = referensi – keluaran *plant* = *set point* – nilai sensor

Untuk persamaan PID no.(4.10) merupakan PID bentuk *independent* dan persamaan (4.11) merupakan PID bentuk *dependent*. Istilah tersebut mengacu kepada ketergantungan setiap suku persamaan terhadap nilai K_p . Untuk persamaan (4.10), jika dilakukan perubahan nilai pada konstanta proporsional (K_p) maka tidak akan mempengaruhi konstanta parameter lainnya. Sedangkan untuk persamaan (4.11), dengan merubah nilai K_p maka akan merubah nilai dari parameter-parameter lainnya. Disini akan digunakan persamaan PID bentuk *independent*. Jika ingin menggunakan persamaan *dependent*, maka tinggal memasukan nilai dari $K_i = K_p/T_i$ dan $K_d = K_p.T_d$

Persamaan (4.7) sampai pada persamaan (4.11) merupakan persamaan dalam kawasan waktu *continuous* (analog). Sedangkan agar persamaan - persamaan tersebut dapat direalisasikan dalam bentuk pemrograman, maka persamaan dalam kawasan waktu *continuous* tersebut harus didiskritisasi terlebih dahulu (kawasan *digital*). Berikut adalah diskritisasi menggunakan metode numerik *rectangular* mundur (*backward rectangular*).

Kontrol Proporsional

Jika dari persamaan (4.7) didiskritisasi maka akan menjadi:

$$u(k) = K_p e(k) \quad (4.12)$$

Berikut ini contoh *list* program dalam mikrokontroler yang menunjukkan kendali proporsional:

```
//kendali proporsional
```

```
error=set_point-nilai_sensor;
```

$outP = K_p * error;$ //nilai K_p ditentukan melalui tuning

Kontrol

Integral

Jika dari persamaan (4.8) didiskritisasi maka akan menjadi:

$$\begin{aligned} u(k) &= K_i \sum_{i=0}^k e(i) T_c \\ u(k) &= K_i T_c \sum_{i=0}^k e(i) = K_i T_c [e(0) + e(1) + \dots + e(k-1) + e(k)] \\ u(k) &= K_i T_c [e(k-1) + e(k)] \end{aligned} \quad (4.13)$$

T_c = waktu sampling atau waktu cuplik (*Sampling time*)

Berikut ini contoh *list* program dalam mikrokontroler yang menunjukkan kendali integral:

```
//kendali integral
errorI=error+error_sbImI;
outI=Ki*errorI*Tc; //nilai Ki ditentukan melalui tuning
error_sbImI=errorI;
```

Kontrol Derivatif

Jika dari persamaan (4.9) didiskritisasi dengan menggunakan cara yang sama seperti kontrol integral maka akan menjadi:

$$u(k) = K_d \frac{e(k) - e(k-1)}{T_c} \quad (4.14)$$

T_c = waktu sampling atau waktu cuplik (*Sampling time*)

Berikut ini contoh *list* program dalam mikrokontroler yang menunjukkan kendali derivatif:

```
//kendali Deferensial
```



```
errorD=error-error_sbImD;
```

```
outD=(Kd*errorD)/Tc; //nilai Kd ditentukan melalui tuning
```

```
error_sbImD=error;
```

Terakhir, Berikut ini contoh *list* program dalam mikrokontroler yang menunjukkan kendali PID:

```
//kontrol PID, merupakan gabungan dari kontrol P, I dan D
```

```
outPID=outP+outI+outD;
```

Agar lebih mengerti dari persamaan yang telah dipaparkan, berikut penjelasannya:

Derivatif (de/dt) adalah suatu operator matematis dalam kawasan *continuous* jika didiskritisasi maka akan menjadi limit, yang merupakan operator matematis dalam kawasan diskrit. Di mana fungsi dari operator limit adalah mengurangi nilai ke k dengan nilai ke k-1. Berdasarkan perhitungan di atas variabel *error* (e) yang di derivatikan, atau dengan kata lain ***error yang sekarang dikurangi error yang sebelumnya***.

Waktu *sampling* adalah lamanya waktu yang digunakan untuk mencuplik atau men *sampling* nilai dari sensor. Nilai dari sensor ini berguna untuk mendapatkan sinyal *error* ($error(e)=set\ point - \text{nilai sensor}$). Di mana waktu *sampling* ini sangat berpengaruh pada kesensitifan sistem yang akan dikontrol.

BAB V

PENGUJIAN DAN ANALISIS

Pengujian dan analisis dilakukan untuk mengetahui apakah sistem telah bekerja sesuai perancangan yang telah dilakukan. Pengujian dilakukan per blok sistem kemudian secara keseluruhan. Pengujian per blok dilakukan untuk mempermudah analisis apabila alat tidak bekerja sesuai dengan perencanaan. Adapun pengujian yang dilakukan sebagai berikut:

- Pengujian catu daya sistem
- Pengujian sensor *rotary encoder*
- Pengujian *driver* motor
- Pengujian rangkaian mikrokontroler ATmega8535
- Pengujian keseluruhan system:
 - Pengujian pengontrolan pemutar gerabah tanpa beban.
 - Pengujian pengontrolan pemutar geraban berbeban.

5.1 Pengujian Catu Daya Sistem

Pengujian rangkaian catu daya sistem bertujuan untuk mengetahui apakah catu daya dari sumber sudah sesuai dengan kebutuhan motor dan kebutuhan rangkaian. Pengujian dilakukan dengan menghubungkan masukan rangkaian catu daya dengan *power supply* dan keluarannya diukur dengan multimeter yang difungsikan sebagai *voltmeter* untuk mengetahui besar tegangan keluaran rangkaian. Hasil pengujian ditunjukkan dalam Gambar 5.1, Gambar 5.2, dan Gambar 5.3.



Gambar 5.1. Hasil Pengujian Catu Daya Menggunakan Multimeter untuk *driver motor*



Gambar 5.2 Hasil Pengujian Catu Daya Menggunakan Multimeter untuk mengaktifkan *coil relay*



Gambar 5.3 Hasil Pengujian Catu Daya Menggunakan Multimeter untuk rangkaian mikrokontroler

Dalam Gambar 5.1 dapat dilihat dari hasil pengujian sumber catu daya adalah 14.6 V, tegangan tersebut sudah cukup untuk menggerakkan Motor DC. Dalam Gambar 5.2 dapat dilihat dari hasil pengujian sumber catu daya adalah 11.72, tegangan tersebut sudah cukup untuk mengaktifkan *coil relay*. Dan juga dari hasil pengujian rangkaian catu daya dalam Gambar 5.3 dapat dilihat bahwa pengukuran nilai tegangan keluaran rangkaian dengan multimeter adalah 5.01 V, tegangan ini masih berada di dalam *range* tegangan kerja dari mikrokontroler berdasarkan datasheet ATmega8535, sehingga dapat diaplikasikan pada sistem.

5.2 Pengujian Sensor Rotary Encoder

Pengujian ini bertujuan untuk mengetahui berfungsi atau tidaknya sensor *Rotary Encoder* dalam mendeteksi putaran. Pengujian ini dapat dilakukan dengan cara mengukur besar tegangan keluaran rangkaian sensor *Rotary Encoder* ketika sensor *Rotary Encoder* terhalang piringan berpola dan ketika tidak terhalang piringan berpola. Hasil pengujian ditunjukkan dalam Tabel 5.1.

Tabel 5.1. Hasil Pengukuran Tegangan Keluaran Rangkaian Sensor *Rotary Encoder*

Percobaan	Terhalang	Tidak Terhalang
1	0.72 V	4.71 V
2	0.91 V	4.74 V
3	0.87 V	4.68 V
4	0.83 V	4.71 V
5	0.86 V	4.73 V
	Rata - rata : 0.83 V	Rata - rata : 4.71 V

Berdasarkan Tabel 5.1 Rata - rata besar tegangan keluaran dari rangkaian sensor *Rotary Encoder* adalah 4.71 V ketika tidak terhalang piringan berpola, hal tersebut telah memenuhi tegangan kerja mikrokontroler ATmega8535 yang berkisar antara 4.5 volt sampai 5.5 volt berdasarkan *datasheet*. Maka ketika terjadi perubahan keadaan pada *Rotary Encoder* saat terhalang dan tidak terhalang, mikrokontroler ATmega8535 akan mendeteksi perubahan logika dari *low* ke *high*. Piringan berpola yang digunakan pada *Rotary Encoder* ini memiliki jumlah 58

pulsa, maka mikrokontroler ATmega8535 akan mendeteksi perubahan logika dari *low* ke *high* sebanyak 58 kali yang nantinya data tersebut akan diolah untuk menentukan besar rpm.

5.3 Pengujian Driver Motor

Pengujian *driver* motor ini bertujuan untuk mengetahui respon dari *driver* dengan masukan dari mikrokontroler dan menguji nilai keluaran dari *driver* terhadap nilai PWM dari mikrokontroler. Pengujiannya adalah melihat nilai *output driver* dengan mengaktifkan *relay driver* dan memberikan nilai PWM pada *driver*. Hasil pengujiannya dapat dilihat dalam Tabel 5.2.

Tabel 5.2 Hasil Pengujian Tegangan *Output Driver* motor terhadap penambahan *duty cycle*

No.	<i>Duty Cycle</i>	Tegangan <i>Output Driver</i> Motor DC (Volt)			
		Percobaan I	Percobaan II	Percobaan III	Rata – rata
1	10%	3.2	3.4	3.4	3.3
2	20%	4.2	4	4.3	4.1
3	30%	5.8	5.5	5.7	5.6
4	40%	7.5	7.3	7.6	7.4
5	50%	8.7	8.7	8.6	8.6
6	60%	10.6	10.3	10.6	10.5
7	70%	11.5	11.5	11.6	11.5
8	80%	13.2	13.1	13.3	13.2
9	90%	14.7	14.6	14.7	14.6
10	100%	14.6	14.5	14.7	14.6

Tabel 5.3 Hasil Pengujian Kecepatan Pemutar Gerabah terhadap Penambahan *duty cycle*

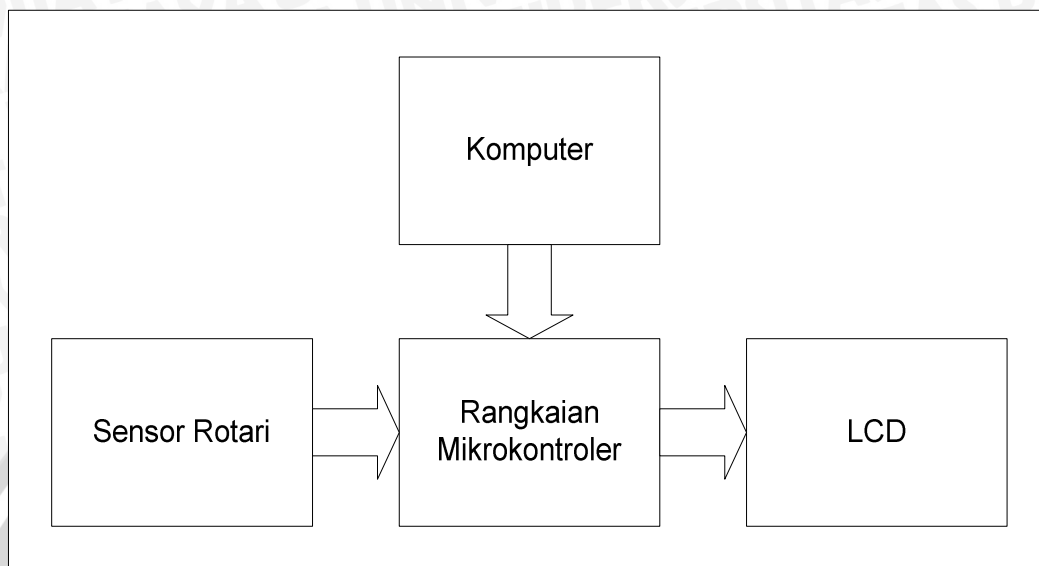
No.	Duty Cycle	Kecepatan Pemutar Gerabah (RPM)			
		Percobaan I	Percobaan II	Percobaan III	Rata – rata
1	10%	0	0	0	0
2	20%	0	0	0	0
3	30%	0	0	0	0
4	40%	18	21	20	19.6
5	50%	49	48	45	47.3
6	60%	78	82	81	80.3
7	70%	98	97	97	97.3
8	80%	118	119	119	118.6
9	90%	133	128	131	130.6
10	100%	129	132	130	130.3

Dari tabel 5.2 dan 5.3 dapat dilihat bahwa pemutar gerabah mulai bekerja pada *duty cycle* 40%, output tegangan *driver* motor pada *duty cycle* 40% adalah 7.4 Volt. Dari tabel 5.2 dapat dilihat pula bahwa kecepatan tertinggi dari pemutar gerabah ini adalah 130 rpm.

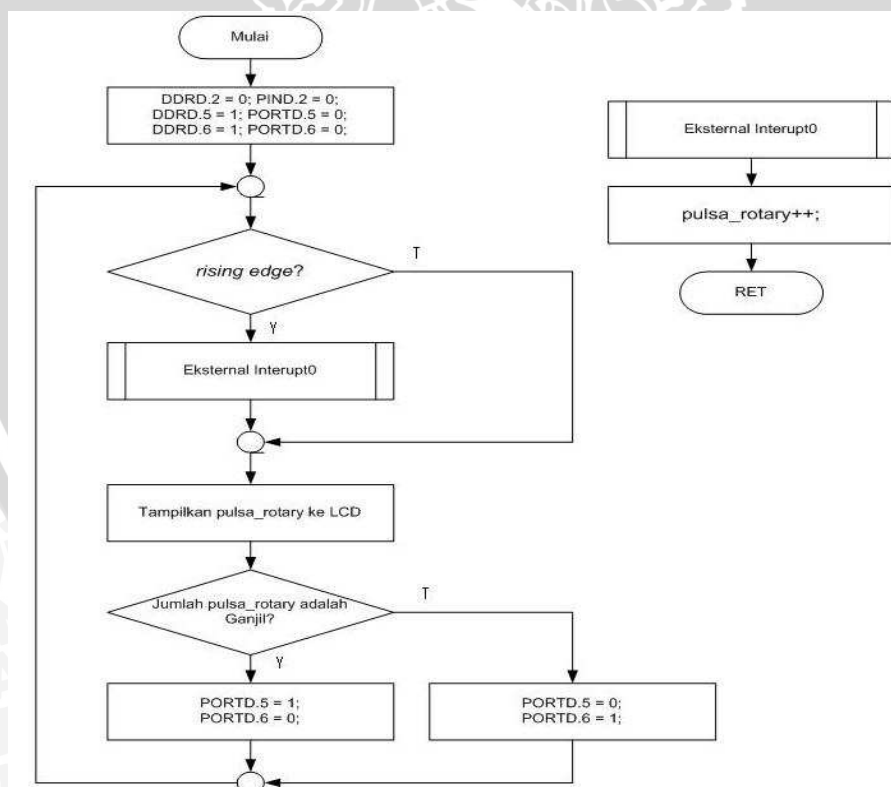
5.4 Pengujian rangkaian mikrokontroler ATmega8535

Pada pemutar gerabah skripsi ini yang digunakan pada mikrokontroler ATmega8535 adalah PIND.2 sebagai input dari sensor *rotary encoder*, PORTD.5 sebagai *output* untuk PWM, dan PORTD.6 sebagai *output* untuk mengaktifkan *coil relay*. Pengujian rangkaian mikrokontroler ini bertujuan untuk mengetahui apakah PORT I/O dan proses aritmatika dalam mikrokontroler dapat berfungsi dengan baik atau tidak. Cara pengujian rangkaian mikrokontroler ini adalah dengan cara *input* (masukan) dihubungkan dengan sensor rotari, sedangkan PORTD.5 dan PORTD.6 digunakan sebagai *output* (keluaran). Ketika terjadi perubahan logika dari 0 ke 1 (*rising edge*) pada PIND.2, sebuah variabel bertipe *integer* bernama *pulsa_rotary* akan bertambah satu nilainya. Kemudian nilai ini akan dilihat bernilai ganjil atau genap, jika bernilai ganjil maka PORTD.5 akan di *set* menjadi 1 dan PORTD.6 akan di *set* menjadi 0, begitu juga sebaliknya. Prosedur pengujian

rangkaian mikrokontroler ini mengikuti diagram blok dalam Gambar 5.4, sedangkan diagram alir program pengujian ditunjukkan dalam Gambar 5.5.



Gambar 5.4. Diagram Blok Pengujian Rangkaian Mikrokontroler



Gambar 5.5. Diagram Alir Program Pengujian Rangkaian Mikrokontroler

Tabel 5.4. Hasil Pengujian Rangkaian Mikrokontroler

Uji ke-	Jumlah Pulsa Rotary	PORTD.5	PORTD.6
1	0	0	1
2	1	1	0
3	2	0	1
4	5	1	0
5	7	1	0
6	8	0	1
7	11	1	0
8	12	0	1
9	19	1	0
10	20	0	1

Berdasarkan hasil pengujian yang ditunjukkan dalam Tabel 5.4 diketahui bahwa saat pulsa rotari adalah ganjil maka PORTD.5 adalah 1 dan PORTD.6 adalah 0, begitu juga sebaliknya. Ini menunjukkan bahwa rangkaian mikrokontroler dapat berfungsi dengan baik.

5.5 Pengujian keseluruhan sistem

Pengujian ini bertujuan untuk mengetahui keberhasilan dari keseluruhan sistem yang dirancang. Pada pengujian ini semua sistem digabungkan menjadi satu kesatuan dan menggunakan kontrol PID sebagai sistem kontrolnya. Pengujian dilakukan untuk mengetahui kontrol sistem terhadap beban yang berubah-ubah.

5.5.4 Peralatan pengujian

Peralatan yang dibutuhkan antara lain:

1. Pemutar Gerabah
2. Sensor *rotary encoder*
3. Rangkaian *Driver motor*
4. *Power Supply*
5. Komputer
6. Sistem pengontrol Mikrokontroler ATmega8535
7. *Software CVAVR*

5.5.5 Prosedur pengujian

Prosedur pengujiannya adalah:

1. Membuat program untuk pengaturan pemutar gerabah
2. Mengatur nilai *set point* pada mikrokontroler nilai sebesar 100 rpm
3. Memasukkan nilai parameter Kp, Ki dan Kd
4. *Downloading* program menuju Mikrokontroler
5. Pada pengujian ini dilakukan dengan meletakkan beban yang berbeda - beda pada alat pemutar gerabah.

5.5.3 Hasil Pengujian

Pertama yang harus dilakukan adalah menentukan berat yang diperbolehkan pada pemutar gerabah. Pada *prototype* pemutar gerabah ini beban yang diperbolehkan adalah 400gr, 600gr, dan 800 gr. Selain itu pemutar gerabah juga diuji tanpa beban ataupun gangguan. Gambar 5.6, Gambar 5.7, Gambar 5.8, dan Gambar 5.9 merupakan gambar grafik respon pemutar gerabah.

5.5.3.1 Hasil Pengujian Alat Pemutar Gerabah Tanpa Beban

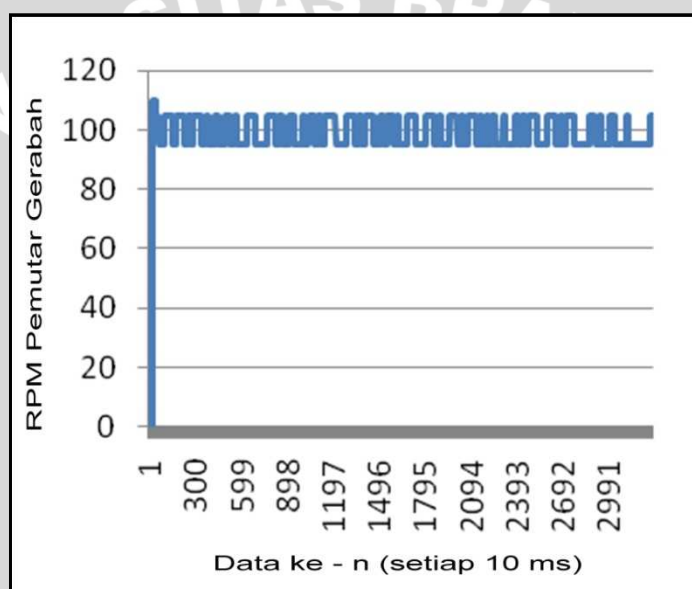
Tabel 5.5. Hasil Pengujian Alat Pemutar Gerabah Tanpa Beban

N0.	Data Ke - n	RPM	No.	Data Ke - n	RPM
1	1	0	71	71	95
2	2	0	72	72	95
3	3	0	73	73	95
4	4	0	74	74	95
5	5	0	75	75	95
6	6	0	76	76	95
7	7	0	77	77	95
8	8	0	78	78	95
9	9	0	79	79	95
10	10	0	80	80	95
11	11	0	81	81	95
12	12	0	82	82	95
13	13	0	83	83	95
14	14	0	84	84	95
15	15	0	85	85	95
16	16	0	86	86	95
17	17	0	87	87	95
18	18	0	88	88	95

19	19	0
20	20	0
21	21	0
22	22	0
23	23	0
24	24	0
25	25	110
26	26	110
27	27	110
28	28	110
29	29	110
30	30	110
31	31	110
32	32	110
33	33	110
34	34	110
35	35	110
36	36	110
37	37	110
38	38	110
39	39	110
40	40	110
41	41	110
42	42	110
43	43	110
44	44	110
45	45	110
46	46	110
47	47	110
48	48	110
49	49	110
50	50	105
51	51	105
52	52	105
53	53	105
54	54	105
55	55	105
56	56	105
57	57	105
58	58	105
59	59	105

89	89	105
90	90	105
91	91	105
92	92	105
93	93	105
94	94	105
95	95	105
96	96	105
97	97	105
98	98	105
99	99	105
100	100	105
101	101	95
102	200	105
103	300	95
104	400	105
105	500	105
106	600	95
107	700	105
108	800	95
109	900	105
110	1000	95
111	1100	105
112	1200	105
113	1300	105
114	1400	95
115	1500	95
116	1600	95
117	1700	95
118	1800	105
119	1900	95
120	2000	95
121	2100	105
122	2200	105
123	2300	105
124	2400	105
125	2500	95
126	2600	95
127	2700	95
128	2800	105
129	2900	105

60	60	105
61	61	105
62	62	105
63	63	105
64	64	105
65	65	105
66	66	105
67	67	105
68	68	105
69	69	105
70	70	105



Gambar 5.6. Grafik Respon Pemutar Gerabah tanpa beban

Gambar 5.6 merupakan gambar grafik yang dihasilkan berdasarkan tabel 5.5. Berdasarkan Tabel 5.5 dan juga terlihat pada gambar 5.6 diketahui bahwa nilai kecepatan sesaatnya adalah 105 RPM dan 95 RPM. Persentase kesalahan yang diperbolehkan dalam pengujian ini adalah 5%. Dengan demikian nilai toleransi yang diperbolehkan adalah sebesar 5 rpm.

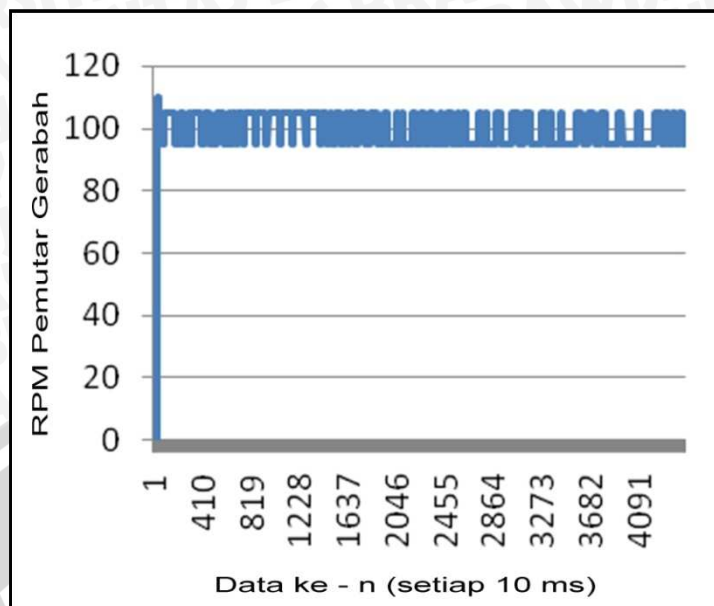
5.5.3.2 Hasil Pengujian Alat Pemutar Gerabah dengan Beban 400gr

Tabel 5.6. Hasil Pengujian Alat Pemutar Gerabah dengan Beban 400gr

N0.	Data Ke - n	RPM	No.	Data Ke - n	RPM
1	1	0	71	71	95
2	2	0	72	72	95
3	3	0	73	73	95
4	4	0	74	74	95
5	5	0	75	75	95
6	6	0	76	76	95
7	7	0	77	77	95
8	8	0	78	78	95
9	9	0	79	79	95
10	10	0	80	80	95
11	11	0	81	81	95
12	12	0	82	82	95
13	13	0	83	83	95
14	14	0	84	84	95
15	15	0	85	85	95
16	16	0	86	86	95
17	17	0	87	87	95
18	18	0	88	88	95
19	19	0	89	89	105
20	20	0	90	90	105
21	21	0	91	91	105
22	22	0	92	92	105
23	23	0	93	93	105
24	24	0	94	94	105
25	25	0	95	95	105
26	26	110	96	96	105
27	27	110	97	97	105
28	28	110	98	98	105
29	29	110	99	99	105
30	30	110	100	100	105
31	31	110	101	200	95
32	32	110	102	300	105
33	33	110	103	400	105
34	34	110	104	500	95
35	35	110	105	600	95
36	36	110	106	700	95
37	37	110	107	800	105

38	38	110
39	39	110
40	40	110
41	41	110
42	42	110
43	43	110
44	44	110
45	45	110
46	46	110
47	47	110
48	48	110
49	49	105
50	50	105
51	51	105
52	52	105
53	53	105
54	54	105
55	55	105
56	56	105
57	57	105
58	58	105
59	59	105
60	60	105
61	61	105
62	62	105
63	63	105
64	64	105
65	65	105
66	66	105
67	67	105
68	68	105
69	69	105
70	70	105

108	900	105
109	1000	105
110	1100	105
111	1200	105
112	1300	105
113	1400	105
114	1500	95
115	1600	95
116	1700	95
117	1800	95
118	1900	95
119	2000	95
120	2100	105
121	2200	105
122	2300	95
123	2400	95
124	2500	105
125	2600	105
126	2700	95
127	2800	105
128	2900	105
129	3000	95
130	3100	95
131	3200	95
132	3300	105
133	3400	95
134	3500	95
135	3600	105
136	3700	105
137	3800	105
138	3900	95
139	4000	95



Gambar 5.7. Grafik Respon Pemutar Gerabah dengan Beban 400 gr

Gambar 5.7 merupakan gambar grafik yang dihasilkan berdasarkan tabel 5.6. Berdasarkan Tabel 5.6 dan juga terlihat pada gambar 5.7 diketahui bahwa nilai kecepatan sesaatnya adalah 105 RPM dan 95 RPM. Persentase kesalahan yang diperbolehkan dalam pengujian ini adalah 5%. Dengan demikian nilai toleransi yang diperbolehkan adalah sebesar 5 rpm.

5.5.3.3 Hasil Pengujian Alat Pemutar Gerabah dengan Beban 600gr

Tabel 5.7. Hasil Pengujian Alat Pemutar Gerabah dengan Beban 600gr

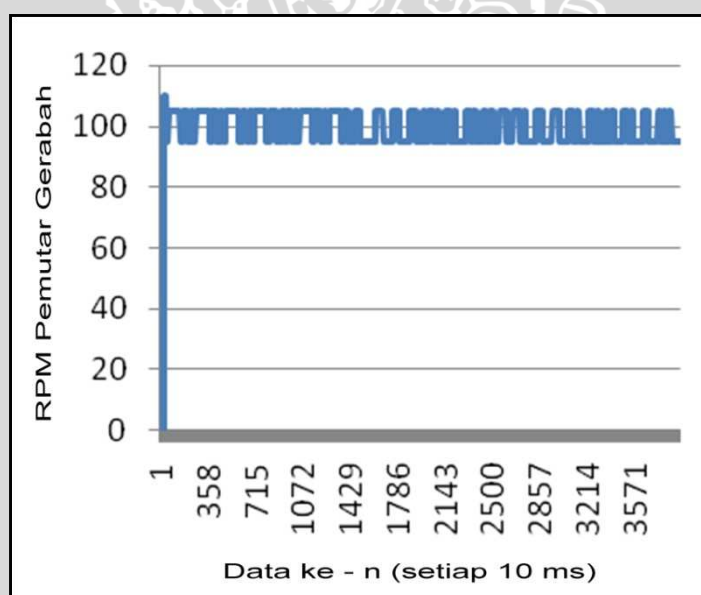
NO.	Data Ke - n	RPM	No.	Data Ke - n	RPM
1	1	0	71	71	105
2	2	0	72	72	105
3	3	0	73	73	105
4	4	0	74	74	105
5	5	0	75	75	105
6	6	0	76	76	105
7	7	0	77	77	105
8	8	0	78	78	105
9	9	0	79	79	105
10	10	0	80	80	105
11	11	0	81	81	105
12	12	0	82	82	105
13	13	0	83	83	105

14	14	0
15	15	0
16	16	0
17	17	0
18	18	0
19	19	0
20	20	0
21	21	0
22	22	0
23	23	110
24	24	110
25	25	110
26	26	110
27	27	110
28	28	110
29	29	110
30	30	110
31	31	110
32	32	110
33	33	110
34	34	110
35	35	110
36	36	110
37	37	110
38	38	110
39	39	110
40	40	110
41	41	110
42	42	110
43	43	110
44	44	110
45	45	110
46	46	110
47	47	95
48	48	95
49	49	95
50	50	95
51	51	95
52	52	95
53	53	95
54	54	95

84	84	105
85	85	105
86	86	105
87	87	105
88	88	105
89	89	105
90	90	105
91	91	105
92	92	105
93	93	105
94	94	105
95	95	105
96	96	105
97	97	105
98	98	105
99	99	105
100	100	105
101	200	105
102	300	105
103	400	105
104	500	105
105	600	95
106	700	95
107	800	105
108	900	95
109	1000	95
110	1100	105
111	1200	105
112	1300	105
113	1400	105
114	1500	95
115	1600	105
116	1700	105
117	1800	95
118	1900	95
119	2000	95
120	2100	105
121	2200	105
122	2300	105
123	2400	95
124	2500	105

55	55	95
56	56	95
57	57	95
58	58	95
59	59	105
60	60	105
61	61	105
62	62	105
63	63	105
64	64	105
65	65	105
66	66	105
67	67	105
68	68	105
69	69	105
70	70	105

125	2600	105
126	2700	95
127	2800	105
128	2900	105
129	3000	95
130	3100	95
131	3200	105
132	3300	105
133	3400	95
134	3500	105



Gambar 5.8. Grafik Respon Pemutar Gerabah dengan Beban 600 gr

Gambar 5.8 merupakan gambar grafik yang dihasilkan berdasarkan tabel 5.7. Berdasarkan Tabel 5.7 dan juga terlihat pada gambar 5.8 diketahui bahwa nilai kecepatan sesaatnya adalah 105 RPM dan 95 RPM. Persentase kesalahan yang diperbolehkan dalam pengujian ini adalah 5%. Dengan demikian nilai toleransi yang diperbolehkan adalah sebesar 5 rpm.

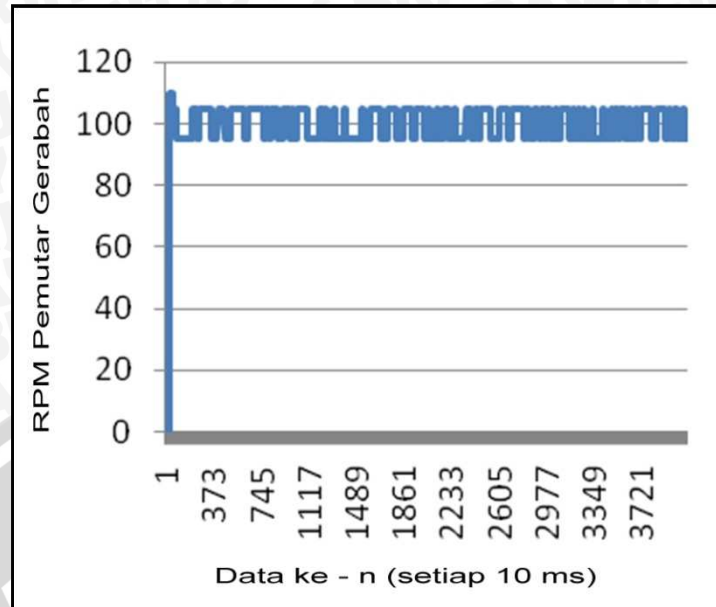
5.5.3.4 Hasil Pengujian Alat Pemutar Gerabah dengan Beban 800gr

Tabel 5.8. Hasil Pengujian Alat Pemutar Gerabah dengan Beban 800gr

NO.	Data Ke - n	RPM	No.	Data Ke - n	RPM
1	1	0	71	71	95
2	2	0	72	72	95
3	3	0	73	73	95
4	4	0	74	74	95
5	5	0	75	75	95
6	6	0	76	76	95
7	7	0	77	77	95
8	8	0	78	78	95
9	9	0	79	79	95
10	10	0	80	80	95
11	11	0	81	81	95
12	12	0	82	82	95
13	13	0	83	83	95
14	14	0	84	84	95
15	15	0	85	85	95
16	16	0	86	86	95
17	17	0	87	87	95
18	18	0	88	88	95
19	19	0	89	89	95
20	20	0	90	90	95
21	21	0	91	91	95
22	22	0	92	92	95
23	23	0	93	93	95
24	24	110	94	94	95
25	25	110	95	95	95
26	26	110	96	96	95
27	27	110	97	97	95
28	28	110	98	98	95
29	29	110	99	99	95
30	30	110	100	100	95
31	31	110	101	101	95
32	32	110	102	200	105
33	33	110	103	300	105
34	34	110	104	400	95
35	35	110	105	500	95
36	36	110	106	600	95

37	37	110
38	38	110
39	39	110
40	40	110
41	41	110
42	42	110
43	43	110
44	44	110
45	45	110
46	46	110
47	47	110
48	48	110
49	49	110
50	50	105
51	51	105
52	52	105
53	53	105
54	54	105
55	55	105
56	56	105
57	57	105
58	58	105
59	59	105
60	60	105
61	61	105
62	62	95
63	63	95
64	64	95
65	65	95
66	66	95
67	67	95
68	68	95
69	69	95
70	70	95

107	700	105
108	800	105
109	900	105
110	1000	105
111	1100	105
112	1200	105
113	1300	105
114	1400	95
115	1500	95
116	1600	95
117	1700	95
118	1800	95
119	1900	95
120	2000	105
121	2100	105
122	2200	95
123	2300	95
124	2400	105
125	2500	105
126	2600	95
127	2700	105
128	2800	95
129	2900	105
130	3000	95
131	3100	95
132	3200	105
133	3300	95
134	3400	105
135	3500	95
136	3600	95
137	3700	105



Gambar 5.9. Grafik Respon Pemutar Gerabah dengan Beban 800 gr

Gambar 5.9 merupakan gambar grafik yang dihasilkan berdasarkan tabel 5.8. Berdasarkan Tabel 5.8 dan juga terlihat pada gambar 5.9 diketahui bahwa nilai kecepatan sesaatnya adalah 105 RPM dan 95 RPM. Persentase kesalahan yang diperbolehkan dalam pengujian ini adalah 5%. Dengan demikian nilai toleransi yang diperbolehkan adalah sebesar 5 rpm.

BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan hasil pengujian tiap blok dan pengujian sistem secara keseluruhan yang telah dilakukan dalam Bab V, dapat disimpulkan beberapa hal sebagai berikut:

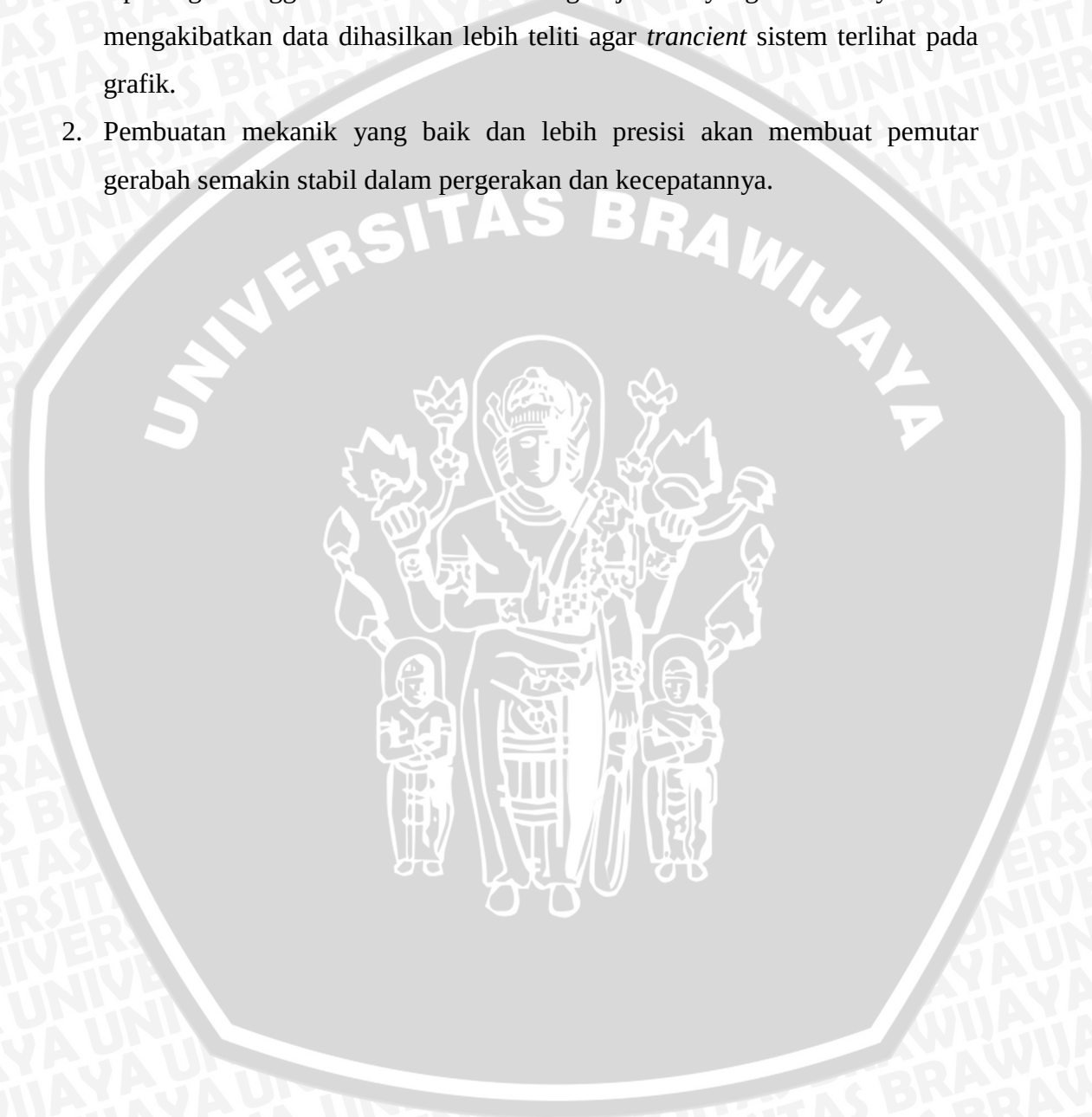
1. Mikrokontroler ATmega8535 dengan menggunakan kontroler PID yang telah dirancang dan diimplementasikan dapat mengatur kecepatan putaran pemutar gerabah. Pemutar gerabah mampu berputar dengan kecepatan yang stabil dengan toleransi kesalahan ± 5 rpm baik berbeban maupun tanpa beban dengan $K_p = 1.5$, $K_i = 2$, dan $K_d = 0.5$.
2. Sistem elektronika yang telah dirancang dapat berfungsi dengan baik, hal tersebut ditunjukkan oleh:
 - Pengujian sensor putaran (*rotary encoder*) saat terhalang piringan berpola rata - rata adalah 0.83 V dan ketika tidak terhalang piringan berpola rata - rata adalah 4.71 V. Maka saat terjadi perubahan keadaan ketika *rotary encoder* terhalang piringan berpola menjadi tidak terhalang, mikrokontroler akan mendeteksi perubahan logika
 - Pemberian besar *duty cycle* yang berbeda berpengaruh kepada *output* tegangan *driver* motor. Pemutar gerabah berputar ketika *output* tegangan *driver* motor sebesar 7.4 V, yaitu ketika *duty cycle* sebesar 40%. dan Pemutar gerabah mempunyai kecepatan tertinggi sebesar 130 RPM. Hal tersebut menunjukkan bahwa *driver* motor dapat berfungsi dengan baik.
 - Ketika dilakukan pengujian mikrokontroler, terjadi kesesuaian antara *input* yang diberikan dengan *output* yang dihasilkan. Hal tersebut menunjukkan bahwa rangkaian mikrokontroler dapat berfungsi dengan baik.

Maka sistem elektronika yang telah dirancang dapat berfungsi dengan baik dan menunjang sistem alat pemutar gerabah secara keseluruhan. Hal ini ditunjukkan dengan adanya kesesuaian antara tujuan dengan sistem yang telah dibuat.

6.2 Saran

Beberapa hal yang direkomendasikan untuk pengembangan lebih lanjut adalah:

1. Penggunaan sensor rotari dengan jumlah lubang yang lebih banyak agar kesalahan pengukuran akan semakin kecil dan pengukuran kecepatan lebih tepat lagi. Penggunaan sensor rotari dengan jumlah yang lebih banyak akan mengakibatkan data dihasilkan lebih teliti agar *transient* sistem terlihat pada grafik.
2. Pembuatan mekanik yang baik dan lebih presisi akan membuat pemutar gerabah semakin stabil dalam pergerakan dan kecepatannya.



DAFTAR PUSTAKA

- ATMEL. 2007. *ATMEGA8535/ATMEGA8535L, 8-bit AVR Microcontroller with 8 Kbytes in System Programable Flash*.
- Bishop, O. 2002. *Dasar-Dasar Elektronika*. Jakarta: Erlangga.
- Kurniawan, Dayat. 2009. *ATMega 8 dan Aplikasinya*. Jakarta: PT Elex Media Komputindo.
- Malvino, Albert Paul. 1996. *Prinsip-Prinsip Elektronika*, Edisi ketiga, Alih bahasa: Hanapi Gunawan. Jakarta: Penerbit Erlangga.
- Ogata, Katsuhiko. 1997. *Teknik Kontrol Automatik (Sistem Pengaturan)*. Erlangga. Jakarta.
- Wardhana, Lingga. 2006. *Baca Sendiri Mikrokontroler AVR Seri ATmega8535 Simulasi, Hardware, dan Aplikasi*. Yogyakarta: Andi
- Anonim. 2000. *Percobaan 10. Analog To Digital Converter*. http://lecturer.eepis-its.edu/~prima/elektronika%20digital/elektronika_digital2/petunjuk-praktikum/perc10-ADC.pdf

LAMPIRAN



LAMPIRAN I

FOTO ALAT





Khasbullah Huda
(0710630045)

Gambar alat pemutar gerabah dari atas

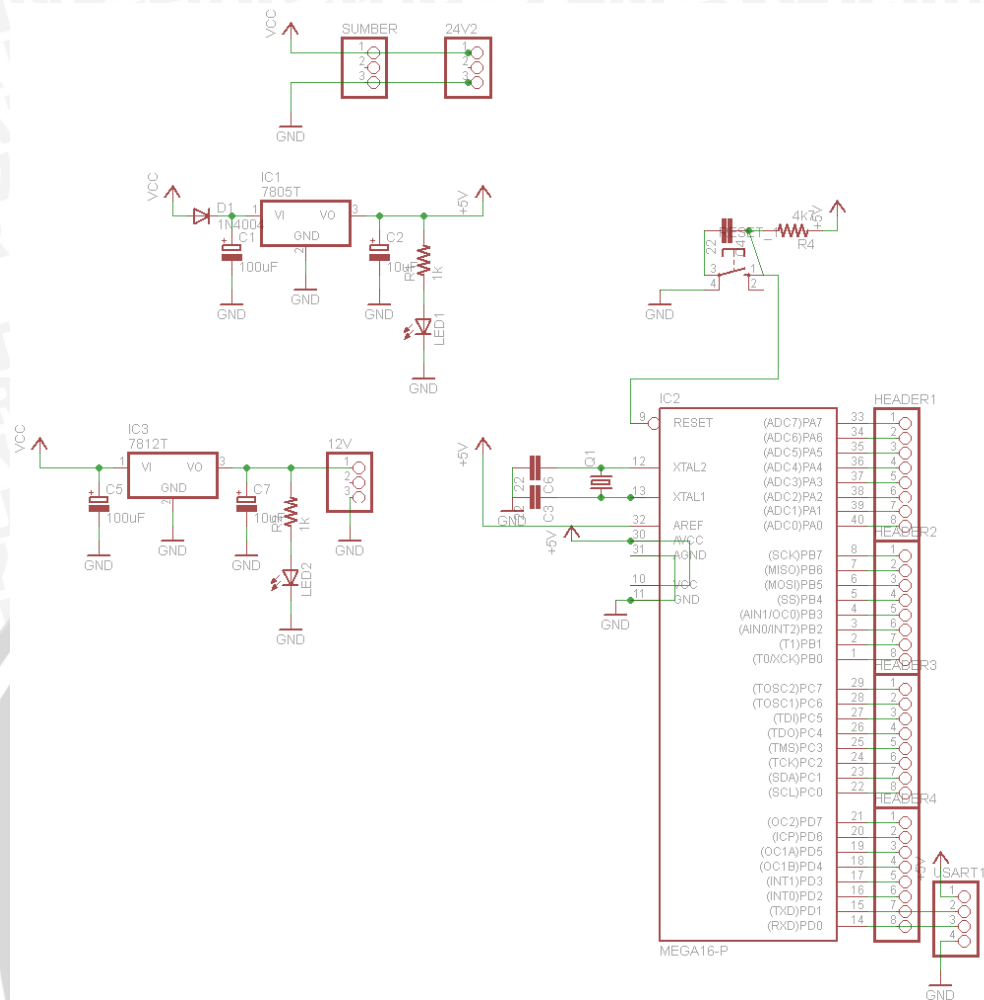


Gambar alat pemutar gerabah dari samping

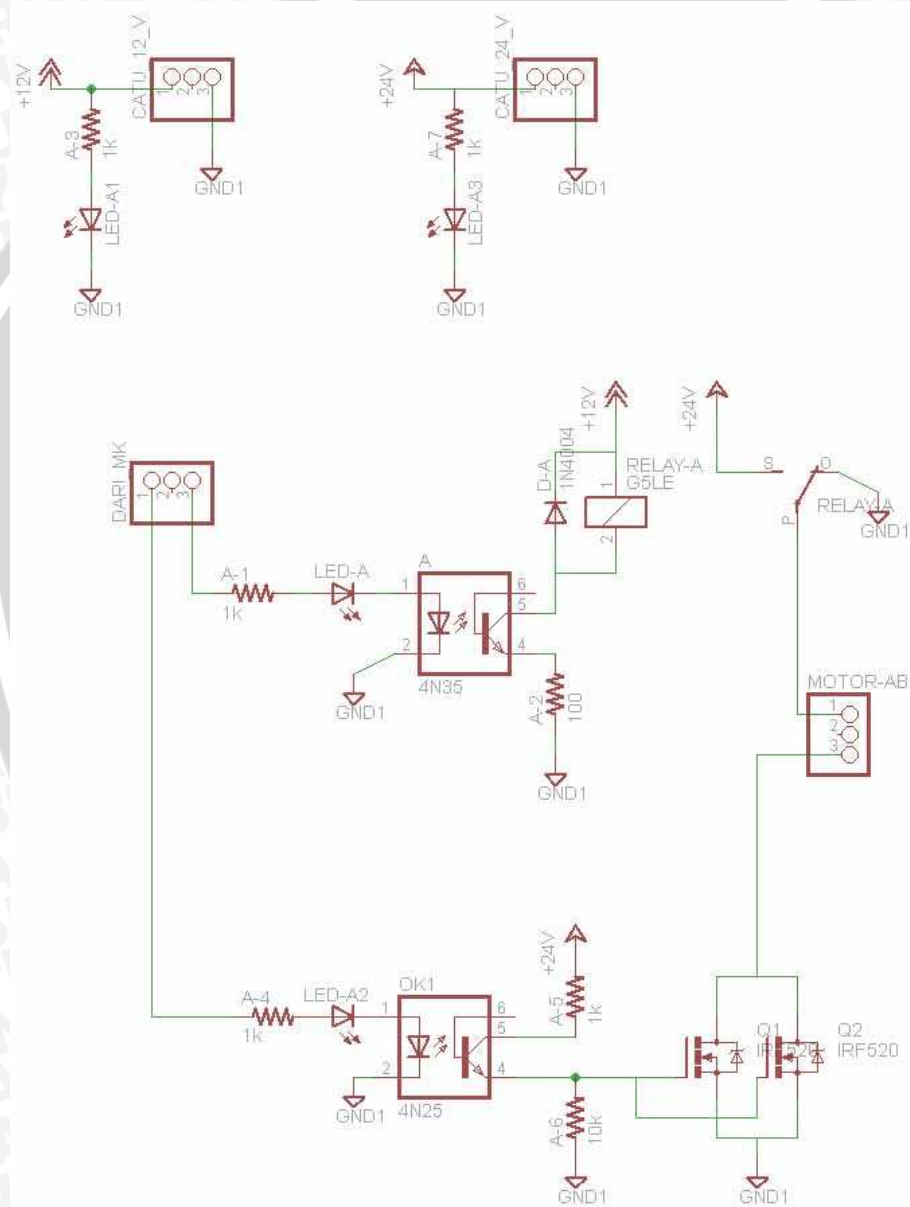
LAMPIRAN II

GAMBAR RANGKAIAN





Rangkaian Minimum Sistem Mikrokontroler



Rangkaian Driver Motor Relay

LAMPIRAN III

LISTING PROGRAM

MIKROKONTROLER ATMEGA8535



```
int pulsa=0;
unsigned char cacah=0;

int rpm;
int rpm2;
int set_point=100;
int Kp=1.5; int Ki=2; int Kd=0.5;
int error_sb1mI=0; int error_sb1mD=0;
int Tc=0.01;
int error; int errorI; int errorD;
int outP; int outI; int outD;
int outPID;
```

```
#include <mega8535.h>
```

```
// Standard Input/Output functions
```

```
#include <stdio.h>
```

```
#include <delay.h>
```

```
// External Interrupt 0 service routine
```

```
interrupt [EXT_INT0] void ext_int0_isr(void)
```

```
{
```

```
// Place your code here
```

```
    pulsa++;
```

```
}
```

```
// Timer 0 overflow interrupt service routine
```

```
interrupt [TIM0_OVF] void timer0_ovf_isr(void)
```

```
{
```

```
    TCNT0=0x8A;           //8A untuk membuat timer 10ms
```

```
    cacah++;
```

```
    if(cacah==10)
```



```
{  
    cacah=0;  
    rpm=(float)pulsa*10*60/58; //formula perhitungan kecepatan  
    pulsa=0;  
}  
}
```

```
void kontrol_PID()
```

```
{  
    //kendali proporsional  
    rpm2=rpm*255/130;  
    error=set_point-rpm2;  
    outP=Kp*error;  
  
    //kendali integral  
    errorI=error+error_sbImI;  
    outI=Ki*errorI*Tc;  
    error_sbImI=errorI;
```

```
    //kendali Derferensial  
    errorD=error-error_sbImD;  
    outD=(Kd*errorD)/Tc;  
    error_sbImD=error;
```

```
    //kontrol PID, merupakan gabungan dari kontrol P, I dan D
```

```
    outPID=outP+outI+outD;
```

```
    if(outPID >= 255)
```

```
    {  
        outPID = 255;  
    }
```

```
else if(outPID <= 0)
```

```
{
```

```
    outPID = 0;
```

```
}
```

```
else outPID=outP+outI+outD;
```

```
}
```

```
void main(void)
```

```
{
```

```
    // Input/Output Ports initialization
```

```
    // Port A initialization
```

```
    // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
```

```
    // State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
```

```
    PORTA=0x00;
```

```
    DDRA=0x00;
```

```
    // Port B initialization
```

```
    // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
```

```
    // State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
```

```
    PORTB=0x00;
```

```
    DDRB=0x00;
```

```
    // Port C initialization
```

```
    // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
```

```
    // State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
```

```
    PORTC=0x00;
```

```
    DDRC=0x00;
```

```
    // Port D initialization
```

```
// Func7=In Func6=Out Func5=Out Func4=In Func3=In Func2=In Func1=In Func0=In
```

```
// State7=T State6=0 State5=0 State4=T State3=T State2=T State1=T State0=T
```

```
PORTD=0x00;
```

```
DDRD=0x60;
```

```
// Timer/Counter 0 initialization
```

```
// Clock source: System Clock
```

```
// Clock value: Timer 0 Stopped
```

```
// Mode: Normal top=FFh
```

```
// OC0 output: Disconnected
```

```
TCCR0=0x05;
```

```
TCNT0=0x8A;
```

```
OCR0=0x00;
```

```
// Timer/Counter 1 initialization
```

```
// Clock source: System Clock
```

```
// Clock value: 16000.000 kHz
```

```
// Mode: Fast PWM top=00FFh
```

```
// OC1A output: Non-Inv.
```

```
// OC1B output: Discon.
```

```
// Noise Canceler: Off
```

```
// Input Capture on Falling Edge
```

```
// Timer 1 Overflow Interrupt: Off
```

```
// Input Capture Interrupt: Off
```

```
// Compare A Match Interrupt: Off
```

```
// Compare B Match Interrupt: Off
```

```
TCCR1A=0x81;
```

```
TCCR1B=0x09;
```

```
TCNT1H=0x00;
```

```
TCNT1L=0x00;
```


ICR1H=0x00;

ICR1L=0x00;

OCR1AH=0x00;

OCR1AL=0x00;

OCR1BH=0x00;

OCR1BL=0x00;

// Timer/Counter 2 initialization

// Clock source: System Clock

// Clock value: Timer 2 Stopped

// Mode: Normal top=FFh

// OC2 output: Disconnected

ASSR=0x00;

TCCR2=0x00;

TCNT2=0x00;

OCR2=0x00;

// External Interrupt(s) initialization

// INT0: On

// INT0 Mode: Falling Edge

// INT1: Off

// INT2: Off

GICR|=0x40;

MCUCR=0x02;

MCUCSR=0x00;

GIFR=0x40;

// Timer(s)/Counter(s) Interrupt(s) initialization

TIMSK=0x01;



```
// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud rate: 9600
UCSRA=0x00;
UCSRB=0x18;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x67;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;

// Global enable interrupts
#asm("sei")
while (1)
{
    // Place your code here
    kontrol_PID();
    OCR1A=outPID;
    printf("%d\r",rpm);
    PORTD.6=1;
};
}
```

LAMPIRAN IV

DATASHEET

