

**SISTEM PENGATURAN KECEPATAN PUTARAN MOTOR  
PADA MESIN PEMUTAR GERABAH MENGGUNAKAN  
KONTROLER PROPORSIONAL INTEGRAL  
DEFERENSIAL (PID) BERBASIS MIKROKONTROLER**

**SKRIPSI**

**JURUSAN TEKNIK ELEKTRO**

Diajukan Untuk Memenuhi Persyaratan  
Memperoleh Gelar Sarjana Teknik



Disusun oleh:

**PRIBADHI HIDAYAT SASTRO**

**NIM. 0810633073 - 63**

**KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN  
UNIVERSITAS BRAWIJAYA  
FAKULTAS TEKNIK  
MALANG  
2013**

**LEMBAR PERSETUJUAN**

**SISTEM PENGATURAN KECEPATAN PUTARAN MOTOR  
PADA MESIN PEMUTAR GERABAH MENGGUNAKAN  
KONTROLER PROPORSIONAL INTEGRAL  
DEFERENSIAL (PID) BERBASIS MIKROKONTROLER**

**SKRIPSI**

**KONSENTRASI TEKNIK KONTROL**

Diajukan untuk memenuhi persyaratan  
memperoleh gelar Sarjana Teknik



Disusun oleh:

**PRIBADHI HIDAYAT SASTRO**

**NIM. 0810633073 - 63**

Telah diperiksa dan disetujui oleh :

**Dosen Pembimbing I**

**Dosen Pembimbing II**

**Ir. Purwanto, MT.**

**NIP. 19540424 198601 1 001**

**Ir. Retnowati, MT.**

**NIP. 19511224 198203 2 001**

**LEMBAR PENGESAHAN**

**SISTEM PENGATURAN KECEPATAN PUTARAN MOTOR  
PADA MESIN PEMUTAR GERABAH MENGGUNAKAN  
KONTROLER PROPORSIONAL INTEGRAL  
DEFERENSIAL (PID) BERBASIS MIKROKONTROLER**

**SKRIPSI**

**JURUSAN TEKNIK ELEKTRO**

Diajukan Untuk Memenuhi Persyaratan  
Memperoleh Gelar Sarjana Teknik

Disusun oleh:

**PRIBADHI HIDAYAT SASTRO**

**NIM. 0810633073 - 63**

Skripsi ini telah diuji dan dinyatakan lulus pada  
Tanggal 2 Agustus 2013

**Majelis Penguji:**

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

**NIP.19741203 200012 1 001**

**Dr. Ir. Erni Yudaningsy, MT.**

**NIP.19650913 199002 2 001**

**Eka Maulana, ST., MT., M.Eng.**

**NIP.841130 06 1 1 0280**

Mengetahui

Ketua Jurusan Teknik Elektro

**Dr. Ir. Sholeh Hadi Pramono, MS.**

**NIP. 19580728 198701 1 001**

## PENGANTAR

Puji dan syukur penulis panjatkan kepada Allah SWT, karena dengan rahmat, taufik dan hidayah-Nya skripsi ini dapat diselesaikan. Skripsi berjudul **“Sistem Pengaturan Kecepatan Putaran Motor Pada Mesin Pemutar Gerabah Menggunakan Kontroler PID Berbasis Mikrokontroler”** ini disusun untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Teknik di Jurusan Teknik Elektro Universitas Brawijaya.

Penulis menyadari bahwa penyusunan skripsi ini tidak terlepas dari bantuan berbagai pihak. Oleh karena itu, dengan ketulusan dan kerendahan hati penulis menyampaikan terima kasih kepada:

1. Allah SWT atas rahmat dan hidayah yang diberikan dan Rasulullah Muhammad SAW semoga sholawat dan salam tetap tercurah kepada beliau,
2. Ayah Sastro Kuncung, ST., W.Eng. dan Ibu Luluk Fujianingsih atas segala nasehat, kasih sayang, perhatian dan kesabarannya di dalam membesarkan dan mendidik penulis, serta telah banyak mendoakan kelancaran penulis hingga terselesaikannya skripsi ini,
3. Prissasova Novelianingsih, adikku yang selalu menemani keseharian dan memberi keceriaan,
4. Ayu Rizky Widowati, terima kasih atas dukungan dan motivasinya,
5. Semua keluarga besar dan saudara yang sudah mendukung,
6. Bapak Ir. Purwanto, MT. selaku Ketua Kelompok Dosen Keahlian Sistem Kontrol Jurusan Teknik Elektro Universitas Brawijaya dan juga Dosen Pembimbing I atas segala ilmu, bimbingan, nasehat, gagasan, ide, saran, motivasi dan bantuan yang telah diberikan,
7. Ibu Ir. Retnowati, MT. selaku Dosen Pembimbing II atas segala ilmu, bimbingan, nasehat, gagasan, ide, saran, motivasi dan bantuan yang telah diberikan,
8. Bapak Dr. Ir. Sholeh Hadi Pramono, MS. selaku Ketua Jurusan Teknik Elektro

Universitas Brawijaya dan Bapak M. Aziz Muslim, ST., MT., Ph.D. selaku Sekretaris Jurusan Teknik Elektro Universitas Brawijaya,

9. Bapak Panca Mudjirahardjo, ST., MT. dan Bapak Ir. Wijono, MT., Ph.D. selaku Dosen Pembimbing akademik penulis atas segala nasehat dan bimbingan yang telah diberikan,
10. Sahabat Nakojop (Nyok, Agus, Okta, Blontang, Limbad, Tiong, Dodid, Mifta, Dakar, Wiyogo, Umar, Aldy, Bojes, Anas, Arnas, Ceper, Fandy, Mono, Jimbon, Hoyi, Samid, Ape, Saprol, Andika, Ade, Pepi, Rio), Against The Pain (Onne, Halim, Alan, Mawan), Parkiran (Cak Muktar, Pak Lan, Cak Ivan, Rendi, Irwan, Engga, Dita, Candra, Ulin), Cikampek (Arsyad, Fajar, Korep, Kico), Lab malam (Awal, Mudito, Tunggul). Terima kasih terutama pada Bagus, Anas, Arnas, dan Rahma atas bantuannya menyelesaikan skripsi ini,
11. Seluruh teman-teman Concordes, senior dan junior, asisten lab serta semua pihak yang tidak mungkin untuk dicantumkan namanya satu per satu, terima kasih banyak atas bantuan dan dukungannya,

Penulis menyadari bahwa skripsi ini masih belum sempurna, oleh karena itu penulis mengharapkan kritik dan saran yang membangun. Penulis berharap semoga skripsi ini dapat bermanfaat bagi pengembangan ilmu pengetahuan dan teknologi.

Malang 2013

Penulis

## ABSTRAK

**Pribadhi Hidayat Sastro**, Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya, 2013, *Sistem Pengaturan Kecepatan Putaran Motor Pada Mesin Pemutar Gerabah Menggunakan Kontroler Proporsional Integral Defereensial (PID) Berbasis Mikrokontroler*, Dosen Pembimbing: Purwanto, Ir., MT. dan Retnowati, Ir., MT.

Kontroler *Proporsional Integral Defereensial* (PID) merupakan kontroler yang memiliki stabilitas yang baik dengan tingkat *error* dan *overshoot* yang kecil. Alat pemutar gerabah yang dirancang diputar oleh motor arus searah atau *direct current* (DC) yang dikontrol oleh mikrokontroler ATMega 32. Alat pemutar gerabah menggunakan penggerak motor (*driver motor*) menggunakan sistem *Pulse Width Modulation* (PWM). Input setting point dilakukan dengan menekan tombol keypad yang berupa data digital, yang kemudian dikonversi ke tegangan oleh mikrokontroler ATMega 32. Keluaran dari mikrokontroler berupa PWM menjadi masukan untuk penggerak *driver motor*, dimana kecepatan keluaran dari motor akan dideteksi oleh sensor *rotary encoder*. Setelah itu akan terjadi pengulangan proses, sampai kecepatan memenuhi nilai set point.

Pemograman dilakukan menggunakan software CodeVisionAVR. Penentuan hasil parameter kontroler PID ini didapatkan dengan menggunakan metode osilasi Ziegler-Nichols. Metode ini dipilih karena dapat mempersingkat waktu pencarian parameter karena menggunakan rumus-rumus sederhana dan proses trial and error hanya pada pencarian parameter Kp. Hasil parameter kontroler PID diperoleh nilai Kp= 1,8, Ki= 0,9 dan Kd= 0,9. Dengan nilai Kp, Ki dan Kd tersebut sistem dapat berjalan dengan baik dengan mempertahankan kecepatan putaran motor mendekati setpoint.

**Kata Kunci** : Pemutar Gerabah, kontroler PID, *Ziegler-Nichols*

## DAFTAR ISI

<b>PENGANTAR</b> .....	i
<b>ABSTRAK</b> .....	iii
<b>DAFTAR ISI</b> .....	iv
<b>DAFTAR GAMBAR</b> .....	vii
<b>DAFTAR TABEL</b> .....	x
<b>BAB I PENDAHULUAN</b> .....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	2
1.5 Sistematika Penulisan.....	2
<b>BAB II TINJAUAN PUSTAKA</b> .....	4
2.1 Teori Gerabah.....	4
2.1.1 Pengertian Gerabah.....	4
2.1.2 Proses Pembuatan Gerabah.....	5
2.2 Kontroler.....	7
2.3 Kontroler <i>Proporsional Integral Diferensial</i> (PID).....	8
2.3.1 Kontroler Propoesional.....	8
2.3.2 Kontroler Integral.....	9
2.3.3 Kontroler Deferensial.....	10
2.3.4 Kontroler PID.....	11
2.3.5 Metode Tuning PID.....	12
2.4 Motor DC.....	18
2.5 Mikrokontroler ATMEGA.....	19

2.6	<i>Liquid Crystal Display (LCD)</i> .....	22
2.7	Driver Motor .....	23
2.8	Sensor <i>Rotary Encoder</i> .....	24
2.9	<i>Pulse Width Modulation (PWM)</i> .....	26
2.10	<i>Buzzer Alarm</i> .....	26
<b>BAB III METODOLOGI PENELITIAN</b> .....		28
3.1	Perancangan Alat.....	28
3.2	Pembuatan Alat .....	28
3.3	Perancangan dan Penyusunan Perangkat Lunak.....	29
3.4	Pengujian Alat.....	29
3.5	Pengambilan Kesimpulan dan Saran.....	30
<b>BAB IV PERANCANGAN DAN PEMBUATAN ALAT</b> .....		31
4.1	Perancangan Sistem.....	31
4.1.1	Blok Diagram Sistem.....	31
4.2	Perancangan Perangkat Keras.....	32
4.2.1	Spesifikasi Alat.....	33
4.2.2	Rangkaian Catu Daya Sistem.....	33
4.2.3	Sensor <i>Rotary Encoder</i> .....	34
4.2.4	Perancangan Driver Motor.....	35
4.2.5	Rangkaian Mikrokontroler .....	35
4.3	Perancangan Parameter PID .....	37
4.3.1	Tuning Eksperimen .....	38
4.4	Perancangan Perangkat Lunak .....	39
4.4.1	Perancangan Algoritma Utama Kontroler PID.....	39
4.4.2	Perancangan Diagram Alir Program Kontroler PID.....	41
4.4.3	Diagram Alir Subrutin PID Sensor.....	42
4.4.4	Diagram Alir Sistem Keseluruhan.....	43

<b>BAB V PENGUJIAN DAN ANALISIS</b> .....	44
5.1 Pengujian Sensor <i>Rotary Encoder</i> .....	44
5.1.1 Peralatan Pengujian .....	45
5.1.2 Prosedur Pengujian .....	45
5.1.3 Hasil Pengujian.....	45
5.2 Pengujian Motor dan Driver Motor.....	48
5.2.1 Peralatan Pengujian .....	48
5.2.2 Prosedur Pengujian .....	48
5.2.3 Hasil Pengujian.....	49
5.3 Pengujian Komunikasi Serial UART ke PC .....	50
5.4 Pengujian Sistem Secara Keseluruhan .....	52
5.4.1 Peralatan Pengujian .....	52
5.4.2 Prosedur Pengujian .....	52
5.4.2.1 Pengujian Kontroler PID.....	54
5.4.3 Hasil Pengujian .....	58
5.4.3.1 Pengujian Set Point 50 RPM.....	58
5.4.3.2 Pengujian Set Point 100 RPM.....	64
<b>BAB VI PENUTUP</b> .....	72
6.1 Kesimpulan .....	72
6.2 Saran .....	72
<b>DAFTAR PUSTAKA</b> .....	73
<b>LAMPIRAN 1 FOTO ALAT</b> .....	74
<b>LAMPIRAN 2 LISTING PROGRAM</b> .....	76
<b>LAMPIRAN 3 DATA SHEET</b> .....	93

## DAFTAR GAMBAR

Gambar 2.1 Diagram Blok Kontroler Proporsional .....	8
Gambar 2.2 Diagram Blok Kontroler Integral.....	9
Gambar 2.3 Diagram Blok Kontroler Deferenensial .....	10
Gambar 2.4 Diagram Blok Kontroler PID.....	12
Gambar 2.5 Fungsi Waktu Sinyal Keluaran dan Sinyal Masukan Kontroler PID.....	12
Gambar 2.6 Kurva Respon Unit Step yang Menunjukkan 25% Maximum Overshoot .....	13
Gambar 2.7 Respon Plan Terhadap Masukan Berupa Unit Step.....	13
Gambar 2.8 Respon Plan berbentuk S .....	14
Gambar 2.9 Sistem Loop Tertutup dengan Kontroler Proporsional .....	15
Gambar 2.10 Osilasi Berkesinambungan dengan periode $P_{cr}$ .....	16
Gambar 2.11 Kurva Tanggapan Sistem Dengan Karakteristik <i>Transient Respon</i> .....	17
Gambar 2.12 Motor DC.....	19
Gambar 2.13 Konfigurasi Pin ATMEGA 32.....	21
Gambar 2.14 Blok Diagram Internal ATMEGA 32.....	22
Gambar 2.15 LCD Character 2x16 .....	23
Gambar 2.16 Rangkaian Driver Motor.....	24
Gambar 2.17 <i>Rotary Encoder</i> Absolut .....	25
Gambar 2.18 <i>Rotary Encoder</i> Relatif.....	25
Gambar 2.19 Sinyal PWM Secara Umum.....	26
Gambar 2.20 <i>Buzzer Alarm</i> .....	27
Gambar 3.1 Desain Rancangan Perangkat Keras.....	29
Gambar 4.1 Gambar Blok Diagram Sistem.....	31
Gambar 4.2 Gambar Rancangan Mekanik.....	33
Gambar 4.3 Rangkaian Catu Daya Sistem .....	34

Gambar 4.4 Rangkaian Sensor <i>Rotary Encoder</i> .....	34
Gambar 4.5 Letak <i>Optical IC</i> Pada Piringan Berpola .....	35
Gambar 4.6 Rangkaian Driver Motor.....	35
Gambar 4.7 Konfigurasi Kaki I/O dari Mikrokontroler ATMEGA 32 .....	36
Gambar 4.8 Diagram Blok Aplikasi Perancangan Kontroler PID Pada Alat Pemutar Gerabah .....	37
Gambar 4.9 Osilasi Berkesinambungan dengan Periode <i>Pcr</i> .....	38
Gambar 4.10 Perancangan Sistem Ziegler-Nichols Metode ke Dua.....	38
Gambar 4.11 Diagram Alir Program Mikrokontroler .....	41
Gambar 4.12 Diagram Alir Subrutin Proses PID Sensor .....	42
Gambar 4.13 (a) Diagram Alir Proses <i>Tuning</i> Parameter pada Perangkat Penerima, (b) Diagram Alir Proses Berjalannya Robot.....	43
Gambar 5.1 Hasil Pengujian <i>Rotary Encoder</i> Menggunakan Osiloskop .....	46
Gambar 5.2 Grafik Hubungan Antara Rpm dan Counter.....	47
Gambar 5.3 Diagram Blok Pengujian <i>Driver</i> Motor DC .....	48
Gambar 5.4 Diagram Pengujian Komunikasi Serial UART RS232 .....	50
Gambar 5.5 Hasil Pengujian Komunikasi Serial dengan RS-232.....	51
Gambar 5.6 Blok Diagram Pengujian Sistem Secara Keseluruhan .....	53
Gambar 5.7 Pengujian Kecepatan Menggunakan RS 232.....	53
Gambar 5.8 Penggunaan Perangkat <i>writer</i> .....	54
Gambar 5.9 Grafik Respon Kecepatan Kotor Pemutar Gerabah dengan $K_p = 1$ .....	55
Gambar 5.10 Grafik Respon Kecepatan Motor Pemutar Gerabah dengan $K_p = 2$ .....	56
Gambar 5.11 Grafik Respon Kecepatan Motor Pemutar Gerabah dengan $K_p = 3$ .....	56
Gambar 5.12 Grafik Respon Kecepatan Motor Pemutar Gerabah pada Set Point 50 rpm Tanpa Beban.....	58
Gambar 5.13 Grafik Respon Kecepatan Motor Pemutar Gerabah pada Set Point 50 rpm 1kg .....	59

Gambar 5.14 Grafik Respon Kecepatan Motor Pemutar Gerabah pada Set Point 50 rpm 2kg .....	60
Gambar 5.15 Grafik Respon Kecepatan Motor Pemutar Gerabah pada Set Point 50 rpm 3kg .....	61
Gambar 5.16 Grafik Respon Kecepatan Motor Pemutar Gerabah pada Set Point 50 rpm 4kg .....	62
Gambar 5.17 Grafik Respon Kecepatan Motor Pemutar Gerabah pada Set Point 50 rpm 5kg .....	63
Gambar 5.18 Grafik Respon Kecepatan Motor Pemutar Gerabah pada Set Point 100 rpm Tanpa Beban.....	64
Gambar 5.19 Grafik Respon Kecepatan Motor Pemutar Gerabah pada Set Point 100 rpm 1kg .....	65
Gambar 5.20 Grafik Respon Kecepatan Motor Pemutar Gerabah pada Set Point 100 rpm 2kg .....	66
Gambar 5.21 Grafik Respon Kecepatan Motor Pemutar Gerabah pada Set Point 100 rpm 3kg .....	67
Gambar 5.22 Grafik Respon Kecepatan Motor Pemutar Gerabah pada Set Point 100 rpm 4kg .....	68
Gambar 5.23 Grafik Respon Kecepatan Motor Pemutar Gerabah pada Set Point 100 rpm 5kg .....	69

## DAFTAR TABEL

Tabel 2.1 Aturan Penalaan Ziegler-Nichols Berdasarkan Respons Unit Step dari Plan .....	15
Tabel 2.2 Aturan Dasar Ziegler-Nichols Berdasarkan Critical Gain Ker dan Critical Period Pcr .....	16
Tabel 4.1 Fungsi Pin Mikrokontroler .....	36
Tabel 5.1 Hasil Pengujian <i>Counter</i> dan Rpm .....	46
Tabel 5.2 Hasil Pengujian <i>Pulse Width Modulation</i> (PWM) .....	49
Tabel 5.3 Tabel Data Performansi Sistem Set Point 50 rpm .....	70
Tabel 5.4 Tabel Data Performansi Sistem Set Point 100 rpm .....	70
Tabel 5.5 Tabel Data Performansi Sistem Menggunakan Kontrol Logika Fuzzy .....	71



## BAB I PENDAHULUAN

### 1.1 Latar Belakang

Kebutuhan masyarakat sangat beragam sehingga inovasi untuk memenuhi kebutuhan masyarakat terus bermunculan. Salah satunya hal yang bersifat dekoratif sangat banyak digemari oleh masyarakat. Dekoratif interior rumah seperti gerabah yang menjadi hiasan rumah banyak digemari. Berbagai bentuk kerajinan banyak yang bisa dihasilkan oleh gerabah seperti patung, mangkok, piring, gelas, celengan, pot, dan guci. Karena banyaknya permintaan dari konsumen, mengakibatkan pengrajin gerabah dituntut untuk memproduksi kerajinan sebanyak-banyaknya dalam waktu sesingkat mungkin. Dalam pembuatan gerabah penggunaan tangan untuk membentuk dan bantuan kaki manusia sebagai sumber gerakan mengakibatkan putaran yang tidak konstan sehingga diperoleh beberapa masalah seperti hasil akhir yang kurang rapi dan menguras tenaga sehingga tidak memenuhi kualitas dan kuantitas. Dari masalah yang dihadapi oleh pengrajin gerabah tersebut, maka penulis mencoba membuat mesin pemutar gerabah untuk membantu mempermudah produksi gerabah.

Saat ini teknologi kontrol *Proporsional Integral Defereensial* (PID) dikenal sebagai kontroler berumpan balik yang paling sering digunakan dalam dunia teknologi. Karena terbukti dapat memberikan performa kontrol yang baik dan berguna untuk mempercepat reaksi sistem dan menghilangkan *offset*. Pada skripsi ini digunakan kontroler PID yang bertujuan untuk membandingkan dengan penelitian sebelumnya yang dilakukan oleh (Rahma 2012) yang menggunakan kontroler logika fuzzy. Pada penelitian sebelumnya diperlukan waktu yang lama untuk mencapai keadaan *steady state*. Pada skripsi ini digunakan perangkat mikrokontroler dengan menggunakan metode PID ziegler nichols yang bertujuan untuk mempercepat respon sistem. Selain itu juga ditambahkan *buzzer* sebagai alarm serta adanya fitur-fitur *display* dan *keypad* guna mempermudah penggunaan alat pemutar gerabah sehingga mempermudah pengrajin dalam pembuatan gerabah.

## 1.2 Rumusan Masalah

Mengacu pada permasalahan yang diuraikan pada latar belakang, maka rumusan masalah dapat ditekankan pada bagaimana kontrol *Proporsional Integral Deferensial* (PID) digunakan sebagai pengontrol kecepatan pada alat pemutar gerabah dengan menggunakan mikrokontroler ATMEGA 32.

## 1.3 Batasan Masalah

Untuk menekankan pada objek pembahasan yang ada maka pada penelitian ini diberikan batasan masalah sebagai berikut:

1. Mesin gerabah yang dibuat merupakan sebuah model atau *prototype*, yang dirancang sesuai aslinya.
2. Pembahasan ditekankan pada proses pengendalian kecepatan motor beserta monitoringnya, kinerja *driver*, perhitungan matematis dan rangkaian elektronika tidak dibahas mendalam.
3. Gangguan berupa berat beban yang berubah-ubah secara acak dalam batasan yang telah ditentukan.

## 1.4 Tujuan

Menciptakan sistem yang dapat mempertahankan kecepatan putaran motor *direct current* (DC) sebagai penggerak mesin pemutar gerabah saat terjadi gangguan berupa perubahan berat beban material dengan mempertahankan keluaran sistem mendekati *setpoint*.

## 1.5 Sistematika Penulisan

Agar penyusunan laporan skripsi ini dapat mencapai sasaran dan tidak menyimpang dari judul yang telah ditentukan, maka diperlukan sistematika pembahasan yang jelas. Pembahasan dalam skripsi ini secara garis besar adalah sebagai berikut:

### **BAB I**                    **Pendahuluan**

Menjelaskan tentang latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika penulisan.

### **BAB II**                    **Tinjauan Pustaka**

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

### **BAB III**                    **Metodologi**

Menjelaskan tentang metodologi penelitian yang terdiri dari studi literatur, perancangan alat, pembuatan alat, pengujian alat, serta pengambilan kesimpulan dan saran.

#### **BAB IV**

#### **Perancangan dan Pembuatan Alat**

Menjelaskan tentang perancangan dan pembuatan alat yang meliputi prinsip kerja alat, perancangan perangkat keras dan perangkat lunak.

#### **BAB V**

#### **Pengujian dan Analisis**

Menjelaskan tentang pengujian alat dan analisa yang meliputi pengujian bagian blok sistem dan pengujian sistem secara keseluruhan.

#### **BAB VI**

#### **Penutup**

Menjelaskan tentang pengambilan kesimpulan sesuai dengan hasil perealisasiian dan pengujian alat sesuai dengan tujuan dan rumusan masalah, serta pemberian saran untuk pengembangan.



## BAB II

### TINJAUAN PUSTAKA

Untuk memudahkan dalam memahami cara kerja rangkaian maupun dasar-dasar perencanaan sistem yang dibuat, maka perlu adanya penjelasan dan uraian mengenai teori penunjang yang digunakan dalam penelitian ini. Teori penunjang yang akan dijelaskan dalam bab ini adalah:

- Teori Gerabah
- Kontroler
- Kontroler *Proporsional Integral Deferensial* (PID)
- Motor DC
- Mikrokontroler ATMEGA 32
- *Liquid Crystal Display* (LCD)
- Driver Motor
- Sensor *Rotary Encoder*
- *Pulse Width Modulation* (PWM)
- *Buzzer Alarm*

#### 2.1 Teori Gerabah

##### 2.1.1 Pengertian Gerabah

Gerabah adalah bagian dari keramik yang dilihat berdasarkan tingkat kualitas bahannya. Namun masyarakat ada yang mengartikan terpisah antara gerabah dan keramik. Ada pendapat gerabah bukan termasuk keramik, karena benda-benda keramik adalah benda-benda pecah belah permukaannya halus dan mengkilap seperti porselin dalam wujud vas bunga, guci, tegel lantai dan lain-lain. Sedangkan gerabah adalah barang-barang dari tanah liat dalam wujud seperti periuk, belanga, tempat air, dll. Untuk memperjelas hal tersebut dapat ditinjau dari beberapa sumber berikut ini.

Menurut The Concise Colombia Encyclopedia, Copyright ã 1995, 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. Usia keramik tertua dikenal dari zaman Paleolitikum 27.000 tahun lalu. 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.

Beberapa teori lain tentang ditemukannya keramik pertama kali, salah satunya terkenal dengan ‘teori keranjang’. Teori ini menyebutkan pada zaman prasejarah, keranjang anyaman digunakan orang untuk menyimpan bahan makanan. Agar tak bocor keranjang tersebut dilapisi dengan tanah liat di bagian 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 (Rosyidi, 2011)

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.

### 2.1.2 Proses Pembuatan Gerabah

Proses pembuatan gerabah yang dilakukan pengrajin 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 ulet dan 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 desain dapat dihasilkan dari tanah liat. Seberapa banyak tanah liat dan berapa lama waktu yang diperlukan tergantung pada seberapa besar gerabah yang akan dihasilkan, bentuk dan desainnya. Perajin gerabah akan menggunakan untuk kedua tangan membentuk tanah liat dan kedua kaki untuk memutar alat pemutar. Kesamaan gerak dan konsentrasi sangat diperlukan untuk dapat melakukannya. Alat-alat yang digunakan yaitu alat pemutar, 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 atau diglasir sehingga terlihat indah dan menarik sehingga bernilai jual tinggi.

## 2.2 Kontroler

Keberadaan kontroler dalam sebuah sistem kontrol mempunyai kontribusi yang besar terhadap perilaku sistem. Pada prinsipnya hal itu disebabkan oleh tidak dapat diubahnya komponen penyusun sistem tersebut. Artinya, karakteristik plant harus diterima sebagaimana adanya, sehingga perubahan perilaku sistem hanya dapat dilakukan melalui penambahan suatu subsistem, yaitu kontroler.

Salah satu fungsi komponen kontroler adalah mengurangi sinyal kesalahan, yaitu perbedaan antara nilai referensi/nilai yang diinginkan dan nilai aktual. Hal ini

sesuai dengan tujuan sistem kontrol di mana mendapat nilai sinyal keluaran sama dengan nilai yang diinginkan/referensi. Semakin kecil kesalahan yang terjadi, semakin baik kinerja sistem kontrol yang diterapkan.

Apabila perbedaan antara nilai referensi dengan nilai keluaran relatif besar, maka kontroler yang baik seharusnya mampu mengatasi perbedaan ini untuk segera menghasilkan sinyal keluaran untuk mempengaruhi plant. Dengan demikian sistem secara cepat mengubah keluaran plant sampai diperoleh selisih dengan nilai referensi sekecil mungkin.

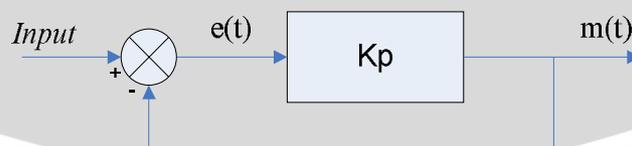
Prinsip kerja kontroler adalah membandingkan nilai aktual keluaran plant dengan nilai referensi, kemudian menentukan nilai kesalahan dan akhirnya menghasilkan sinyal kontrol untuk meminimalkan kesalahan (Ogata, 1995).

### 2.3 Kontroler PID (*Proporsional Integral Deferenisial*)

#### 2.3.1 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 diagram blok 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: 157

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

$$m(t) = K_p e(t) \dots\dots\dots(2-1)$$

Sumber: Ogata, 1995: 157

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 respons dan mengurangi kesalahan keadaan mantap.

**2.3.2 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 mantap 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. Apabila sinyal kesalahan tidak mengalami perubahan, keluaran akan menjaga keadaan seperti sebelum terjadinya perubahan masukan. Gambar 2.2 menunjukkan diagram blok kontroler integral.



**Gambar 2.2** Diagram Blok Kontroler Integral

Sumber: Ogata, 1995: 158

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

$$\frac{dm(t)}{dt} = K_i \cdot e(t) \dots\dots\dots(2-2)$$

$$m(t) = K_i \int_0^t e(t) dt \dots\dots\dots(2-3)$$

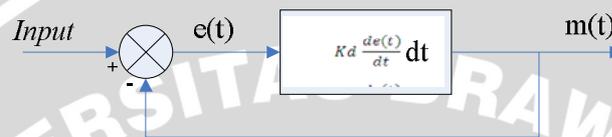
Sumber: Ogata, 1995: 157

dengan  $K_i$  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 kontrol integral digunakan untuk

menghilangkan kesalahan posisi dalam keadaan mantap (*error steady state*) tanpa memperhitungkan kecepatan respons.

### 2.3.3 Kontroler Deferensial

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



Gambar 2.3 Diagram Blok Kontroler Deferensial

Sumber: Ogata, 1995: 177

Nilai keluaran kontroler  $m(t)$  sebanding laju sinyal kesalahan  $\frac{de(t)}{dt}$ . Hubungan ini dapat ditulis sebagai:

$$m(t) = Kd \frac{de(t)}{dt} \dots \dots \dots (2-4)$$

Sumber: Ogata, 1995: 179

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

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

### 2.3.4 Kontroler Proporsional Integral Deferensial (PID)

Setiap kekurangan dan kelebihan dari masing-masing kontroler *Proporsional* (P), *Integral* (I) dan *Deferensial* (D) dapat saling menutupi dengan menggabungkan ketiganya secara paralel menjadi kontroler *proporsional integral deferensial* (PID). Elemen-elemen kontroler P, I dan D masing-masing secara keseluruhan bertujuan untuk mempercepat reaksi sebuah sistem, menghilangkan *offset* dan menghasilkan perubahan

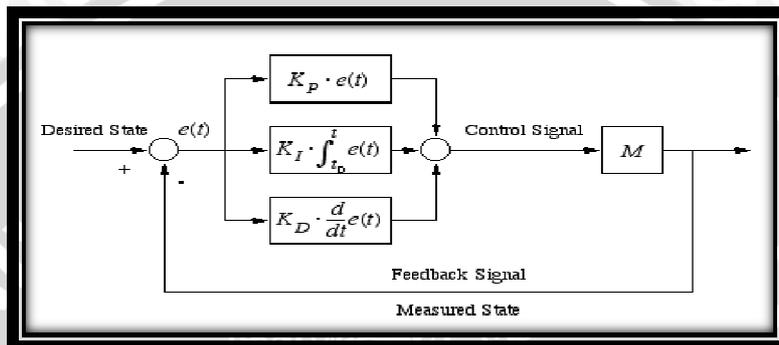
awal yang besar (Gunterus, 1994, 8-10). Kontroler PID memiliki diagram kendali seperti yang ditunjukkan dalam Gambar 2.4.

Aksi kontrolnya dinyatakan sebagai:

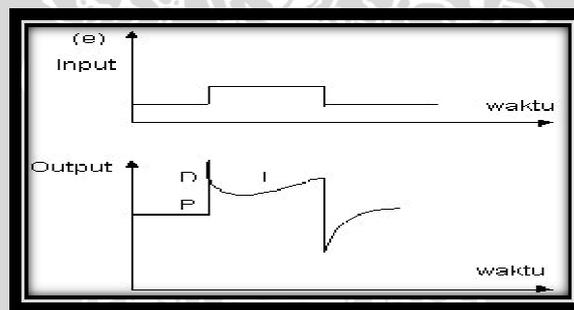
$$m(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt + K_p T_d \frac{de(t)}{dt} \dots \dots \dots (2-5)$$

Sumber: Ogata, 1995: 183

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



Gambar 2.4 Diagram Blok Kontroler PID



Gambar 2.5 Fungsi Waktu Sinyal Keluaran dan Sinyal Masukan Kontroler PID

Sumber: Gunterus, 1994:8-11

Keluaran kontroler PID merupakan penjumlahan dari keluaran kontroler proporsional, integral dan deferensial. Gambar 2.5 menunjukkan hubungan tersebut. Karakteristik kontroler PID sangat dipengaruhi oleh kontribusi besar dari ketiga parameter P,I dan D. Penyetelan konstanta Kp, Ti dan Td 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 kontribusi pada respons sistem secara keseluruhan (Gunterus, 1994, 8-10).

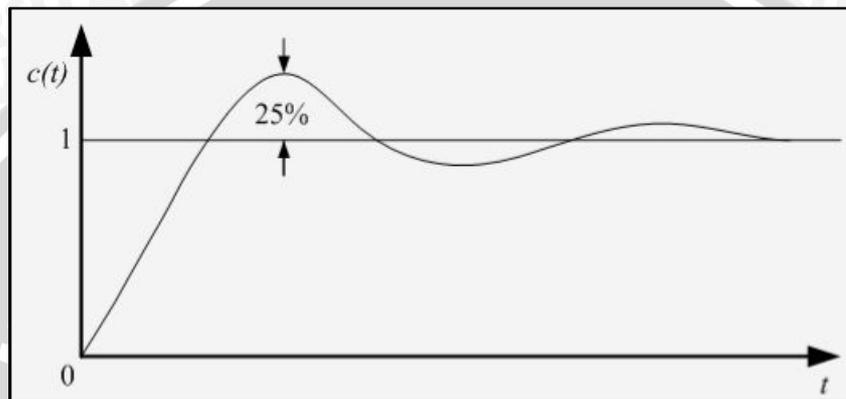
### 2.3.5 Metode Tuning PID

#### Metode Ziegler-Nichols.



Ziegler dan Nichols mengemukakan aturan-aturan untuk menentukan nilai dari gain proporsional  $K_p$ , waktu integral  $T_i$ , dan waktu derivatif  $T_d$  berdasarkan karakteristik respon transien dari plant yang diberikan. Penentuan parameter kontroler PID atau penalaan kontroler PID tersebut dapat dilakukan dengan bereksperimen dengan plant. (Ogata, K., 1997)

Terdapat dua metode yang disebut dengan aturan penalaan Ziegler-Nichols, pada kedua metode tersebut memiliki tujuan yang sama yaitu untuk mencapai 25% maximum overshoot pada respon unit step, seperti ditunjukkan dalam Gambar 2.6.

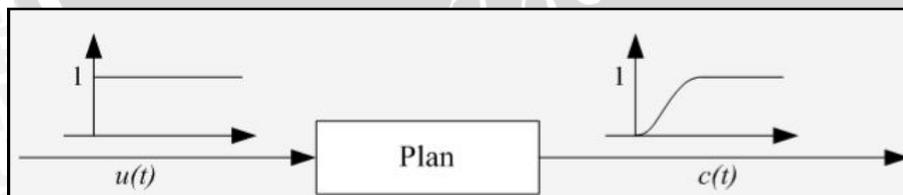


**Gambar 2.6** Kurva Respon Unit Step yang Menunjukkan 25% Maximum Overshoot

Sumber: Teknik Kontrol Automatik, Katsuhiko Ogata, 1997

a) Metode Pertama

Metode pertama atau sering disebut metode kurva reaksi, respon dari plant dapat diperoleh secara eksperimental dengan masukan berupa unit step, seperti yang ditunjukkan dalam Gambar 2.7

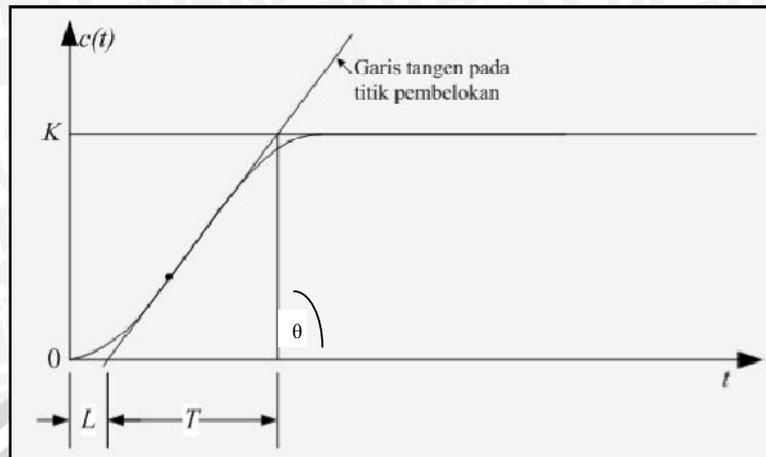


**Gambar 2.7** Respons Plan Terhadap Masukan Berupa Unit Step

Sumber: Teknik Kontrol Automatik, Katsuhiko Ogata, 1997

Jika dalam plan tersebut terdapat integrator atau dominan complex-conjugate poles, maka kurva respons unit step berbentuk seperti huruf S, seperti dalam Gambar

2.8. Jika respons tidak memberikan bentuk kurva S, maka metode ini tidak berlaku. (Ogata, K., 1997).



Gambar 2.8 Respons Plan berbentuk S

Sumber: Teknik Kontrol Automatik, Katsuhiko Ogata, 1997

Kurva berbentuk S tersebut dapat dikarakteristikan menjadi dua konstanta yaitu waktu tunda L dan konstanta waktu T. Waktu tunda dan konstanta waktu ditentukan dengan menggambar sebuah garis tangen pada titik pembelokan dari kurva S, dan menentukan perpotongan antara garis tangen dengan sumbu waktu t dan sumbu  $c(t) = K$ , seperti yang telah ditunjukkan dalam Gambar 2.9 Fungsi alih  $C(s)/U(s)$  dapat dilakukan pendekatan dengan sistem orde satu dengan persamaan sebagai berikut:

$$\frac{c(s)}{U(s)} = \frac{Ke^{-Ls}}{Ts+1} \dots\dots\dots(2-6)$$

Ziegler dan Nichols menyarankan untuk menentukan nilai-nilai dari  $K_p$ ,  $T_i$  dan  $T_d$  berdasarkan pada formula yang ditunjukkan dalam Tabel 2.1 (Ogata, K., 1999)

Tabel 2.1 Aturan Penalaan Ziegler-Nichols Berdasarkan Respons Unit Step Dari Plan

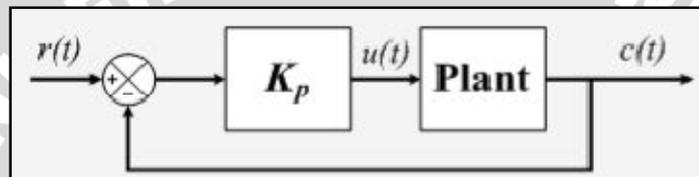
Tipe Kontroler	$K_p$	$T_i$	$T_d$
P	$\frac{T}{L}$	$\infty$	0
PI	$0,9\frac{T}{L}$	$\frac{L}{0,3}$	0
PID	$1,2\frac{T}{L}$	2L	0,5 L

Sumber: Ogata, K. 1997

b) Metode Kedua

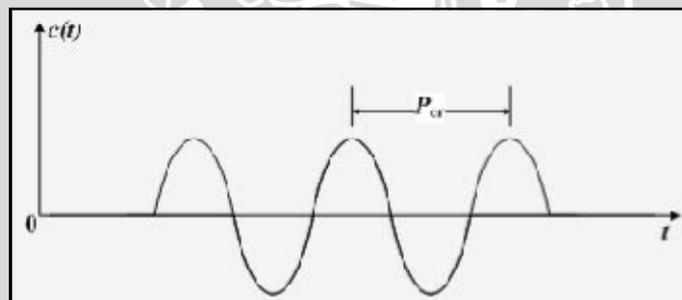


Dalam metode kedua ziegler-nichols, mula-mula yang dilakukan adalah membuat  $T_i = 0$  dan  $T_d = 0$ . Kemudian hanya dengan menggunakan tindakan kontrol proporsional, harga ditingkatkan dari nol ke suatu nilai kritis  $K_{cr}$ , disini mula-mula keluaran memiliki osilasi yang berkesinambungan (Jika keluaran tidak memiliki osilasi berkesinambungan untuk nilai  $K_p$  manapun yang telah diambil, maka metode ini tidak berlaku). Dari keluaran yang berosilasi secara berkesinambungan, penguatan kritis  $K_{cr}$  dan periode  $P_{cr}$  dapat ditentukan. Diagram blok sistem loop tertutup dengan kontroler proporsional dapat dilihat dalam Gambar 2.8 dan untuk osilasi berkesinambungan dengan periode  $P_{cr}$  dapat dilihat dalam Gambar 2.9 Ziegler dan Nichols menyarankan penyetelan nilai parameter  $K_p, T_i, T_d$  dan berdasarkan rumus yang diperlihatkan dalam Tabel 2.2. (Ogata, K., 1997)



**Gambar 2.9** Sistem Loop Tertutup dengan Kontroler Proporsional

Sumber: Teknik Kontrol Automatik, Katsuhiko Ogata, 1997



**Gambar 2.10** Osilasi Berkesinambungan dengan periode  $P_{cr}$

Sumber: Teknik Kontrol Automatik, Katsuhiko Ogata, 1997

**Tabel 2.2** Aturan Dasar Ziegler-Nichols Berdasarkan Critical Gain  $K_{cr}$  dan Critical Period  $P_{cr}$

Tipe Kontroler	$K_p$	$T_i$	$T_d$
P	$0.5 K_{cr}$	$\infty$	0
PI	$0.45 K_{cr}$	$\frac{1}{1.2} P_{cr}$	0
PID	$0.60 K_{cr}$	$0.5 P_{cr}$	$0.125 P_{cr}$

Sumber: Teknik Kontrol Automatik, Katsuhiko Ogata, 1997

Respon transien dalam sistem kontrol praktis sering menunjukkan osilasi teredam sebelum mencapai keadaan stabil. Berikut adalah parameter karakteristik respon transien terhadap masukan *unit step*:

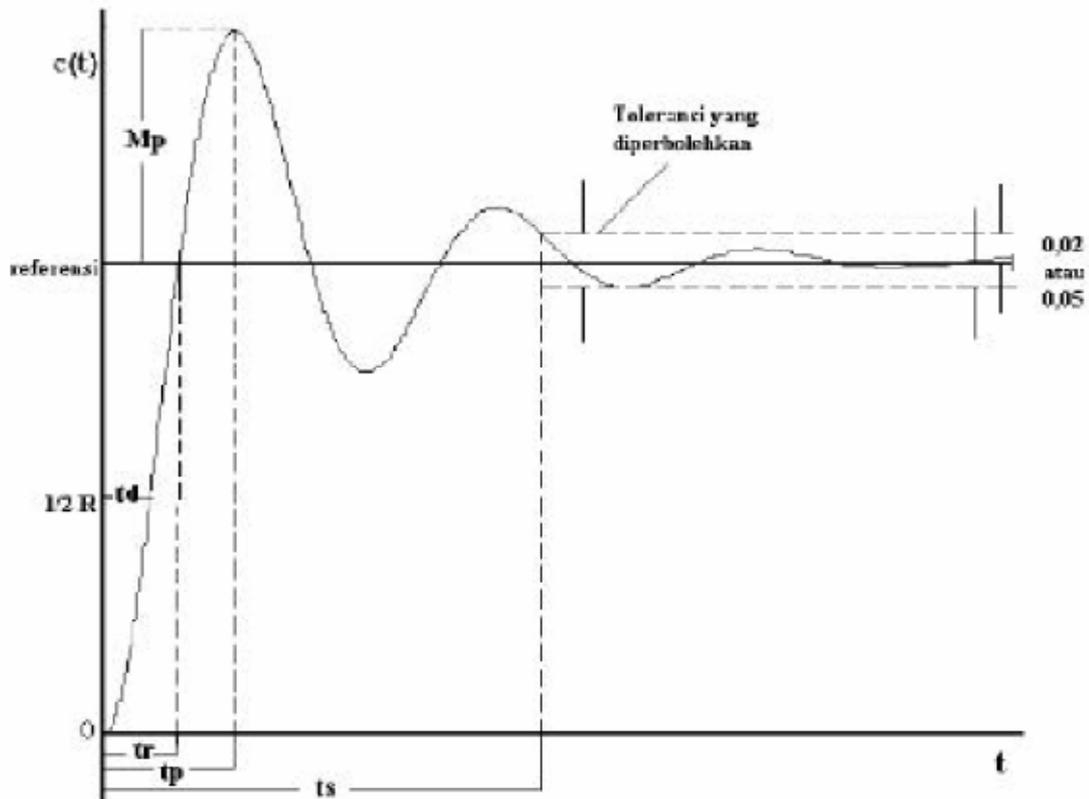
1. *Delay time* ( $t_d$ ) adalah waktu yang diperlukan respon untuk mencapai setengah harga akhir yang pertama kali.
2. *Rise time* ( $t_r$ ) adalah waktu yang diperlukan respon untuk naik dari 10% sampai 90%, 5% sampai 95% atau 0% sampai 100% dari harga akhirnya.
3. *Peak time* ( $t_p$ ) adalah waktu yang diperlukan respon untuk mencapai puncak lewatan yang pertama kali.
4. *Maximum overshoot* ( $M_p$ ) adalah persen harga puncak maksimum dari kurva respon yang diukur dari satu. Jika harga keadaan tunak tidak sama dengan satu, maka biasanya digunakan persen lewatan maksimum yang didefinisikan seperti Persamaan 2.7.

$$M_p = \frac{c(t_p) - c(\infty)}{c(\infty)} \times 100\% \dots\dots\dots(2.7)$$

Besarnya (persen) lewatan maksimum secara langsung menunjukkan kestabilan relative sistem.

5. *Settling time* ( $t_s$ ) adalah waktu yang diperlukan kurva respon untuk mencapai dan menetap dalam daerah disekitar harga akhir yang ukurannya ditentukan dengan presentase mutlak dari harga akhir (biasanya 2% atau 5%).

Berikut merupakan grafik dari respon metode osilasi pada gambar 2.11



Gambar 2.11 Kurva tanggapan sistem dengan karakteristik *transient respon*

## 2.4 Motor DC

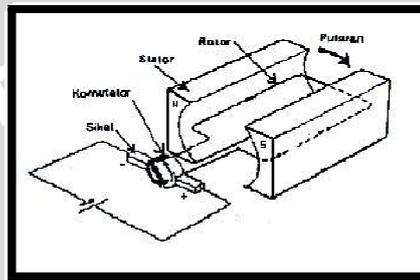
Motor DC yang digunakan dalam penelitian ini adalah Motor DC yang menggunakan magnet permanen. Alasan pemilihan Motor DC tipe ini adalah kemudahan dalam pengontrolan dengan menggunakan pengaturan tegangan DC. Medan stator motor jenis ini dihasilkan oleh magnet permanen bukan elektromagnet. Penggunaan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

Motor DC 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 2.12.



Gambar 2.12 Motor DC

Sumber: Kismet F [1994:98]

## 2.5 Mikrokontroler ATMEGA

### 2.5.1 Pengenalan Mikrokontroler ATMEGA 32

Mikrokontroler AVR (*Alf and Vegard's Risc processor*) memiliki arsitektur RISC 8 Bit, sehingga semua instruksi dikemas dalam kode 16-bit (16-bits *word*) dan sebagian besar instruksi dieksekusi dalam satu siklus instruksi *clock*. Dan ini sangat membedakan sekali dengan instruksi MCS-51 (Berarsitektur CISC (*Complex Instruction Set Computer*)) yang membutuhkan siklus 12 *clock*.

Adapun kelebihan dari mikrontroler adalah sebagai berikut:

1. Penggerak pada mikrontroler menggunakan bahasa pemrograman *assembly* dengan berpatokan pada kaidah digital dasar sehingga pengoperasian sistem menjadi sangat mudah dikerjakan sesuai dengan logika sistem (bahasa *assembly* ini mudah dimengerti karena menggunakan aplikasi dimana parameter *input* dan *output* langsung bisa diakses tanpa menggunakan banyak perintah). Desain bahasa *assembly* ini tidak menggunakan begitu banyak syarat penulisan bahasa pemrograman seperti huruf besar dan huruf kecil untuk bahasa *assembly* tetap diwajibkan.

2. Mikrokontroler tersusun dalam satu *chip* di mana prosesor, memori, dan I/O terintegrasi menjadi satu kesatuan kontrol sistem sehingga mikrokontroler dapat dikatakan sebagai komputer mini yang dapat bekerja secara inovatif sesuai dengan kebutuhan sistem.
3. Sistem *running* bersifat berdiri sendiri tanpa tergantung dengan komputer sedangkan parameter komputer hanya digunakan untuk *download* perintah instruksi atau program. Langkah-langkah untuk *download* komputer dengan mikrokontroler sangat mudah digunakan karena tidak menggunakan banyak perintah.
4. Pada mikrokontroler tersedia fasilitas tambahan untuk pengembangan memori dan I/O yang disesuaikan dengan kebutuhan sistem.
5. Harga untuk memperoleh alat ini lebih murah dan mudah didapat.

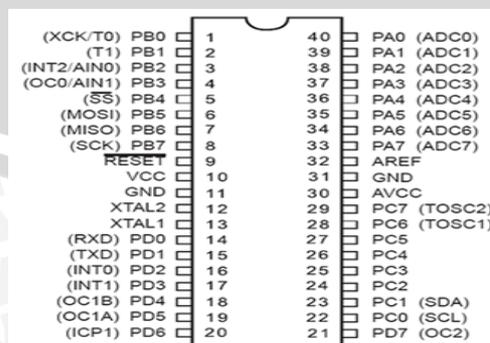
### 2.5.1.1 Fitur ATMEGA 32

Fitur- fitur dalam mikrokontroler ATMEGA 32 meliputi:

- Sistem *processor* 8 bit berbasis *RISC* dengan kecepatan maksimal 16 MHz.
- Ukuran memori *flash* 8KB, *SRAM* sebesar 512 byte, *EEPROM* sebesar 512 byte.
- *ADC (Analog to Digital Converter)* internal dengan resolusi 10 bit sebanyak 8 channel
- Port komunikasi serial *USART* dengan kecepatan maksimal 2.5 Mbps
- Mode *Sleep* untuk penghematan penggunaan daya listrik

### 2.5.1.2 Konfigurasi Pin ATMEGA 32

Konfigurasi Pin mikrokontroler ATMEGA 32 ditampilkan dalam Gambar 2.13.



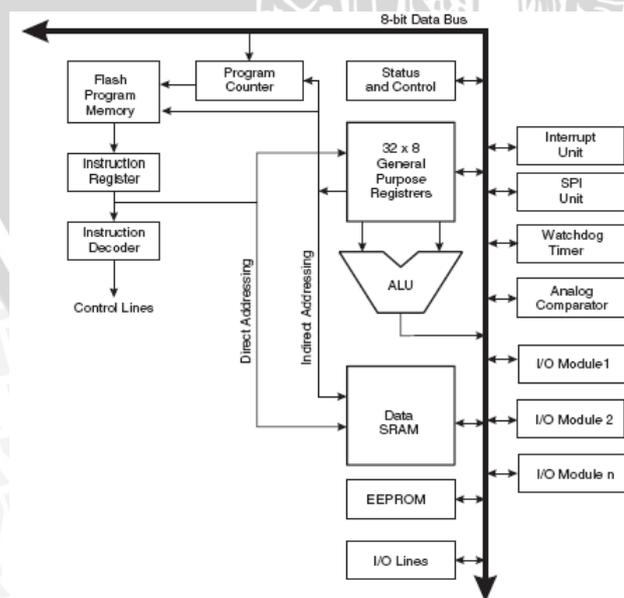
Gambar 2.13 Konfigurasi Pin ATMEGA 32

Sumber: Irwansetyo, 2010

- VCC merupakan Pin yang berfungsi sebagai pin masukan catudaya

- GND merupakan Pin *Ground*
- Port A (PA0...PA7) merupakan pin I/O dan pin masukan ADC
- Port B (PB0...PB7) merupakan pin I/O dan pin yang mempunyai fungsi khusus yaitu *Timer/Counter*, komparator Analog dan SPI (*Serial Port Interface*)
- Port C (PC0...PC7) merupakan port I/O dan pin yang mempunyai fungsi khusus, yaitu komparator analog dan *Timer Oscillator*
- Port D (PD0...PD1) merupakan port I/O dan pin fungsi khusus yaitu komparator analog dan *interrupt* eksternal serta komunikasi serial
- RESET merupakan pin yang digunakan untuk mereset mikrokontroler
- XTAL1 dan XTAL2 merupakan pin masukan *clock* eksternal
- AVCC merupakan pin masukan untuk tegangan ADC
- AREF merupakan pin masukan tegangan referensi untuk ADC

AVR menjalankan sebuah instruksi tunggal dalam satu siklus dan memiliki struktur I/O yang cukup lengkap sehingga penggunaan komponen eksternal dapat dikurangi. Mikrokontroler AVR didesain menggunakan arsitektur *Harvard*, di mana ruang dan jalur bus bagi memori program dipisahkan dengan memori data. Berikut akan ditampilkan blok diagram Internal ATMEGA 32, seperti dalam Gambar 2.14.

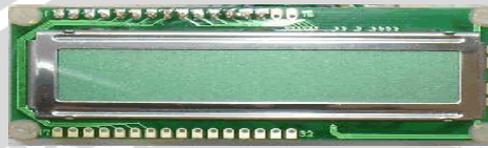


**Gambar 2.14** Blok Diagram Internal ATMEGA 32

Sumber: Irwansetyo, 2010

## 2.6 Liquid Crystal Display (LCD)

LCD (*Liquid cristal display*) adalah salah satu komponen elektronika yang berfungsi sebagai tampilan suatu data, baik karakter, huruf ataupun grafik. Jenis LCD yang dipakai pada alat ini adalah LCD M1632, seperti yang tertera dalam Gambar 2.3. LCD terdiri dari dua bagian, yang pertama merupakan panel LCD sebagai media penampil informasi dalam bentuk huruf/ angka dua baris, masing – masing baris bisa menampung 16 huruf/ angka. Berikut merupakan tampilan dari LCD pada Gambar 2.15



**Gambar 2.15** LCD Character 2x16

Sumber: Irwansetyo, 2010

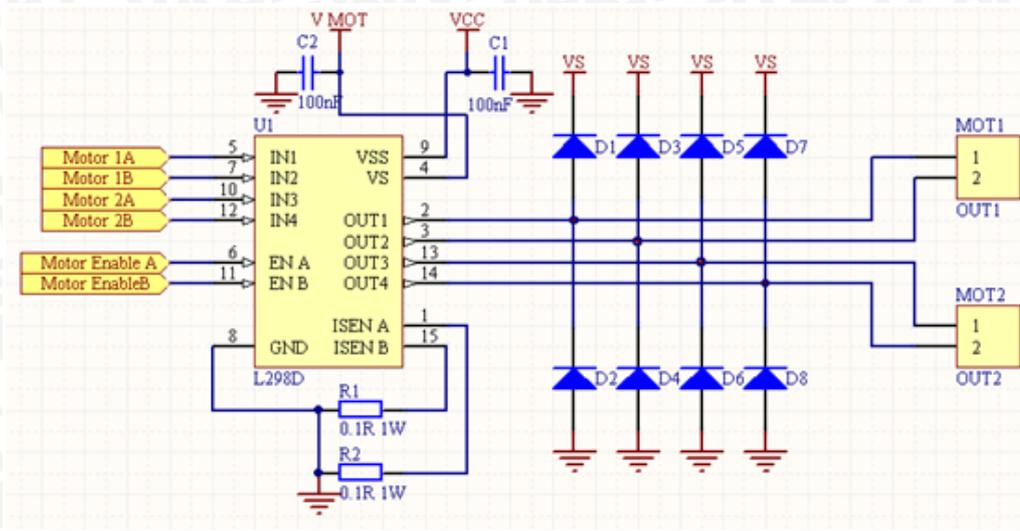
Bagian kedua merupakan sebuah sistem yang dibentuk dengan mikrokontroler yang ditempel dibalik pada panel LCD, berfungsi mengatur tampilan LCD. Dengan demikian pemakaian LCD M1632 menjadi sederhana, sistem lain cukup mengirimkan kode – kode ASCII dari informasi yang ditampilkan. Berikut merupakan spesifikasi LCD M1632 :

1. Tampilan 16 karakter 2 baris dengan matrik 5 x 7 + kursor.
2. ROM pembangkit karakter 192 jenis.
3. RAM pembangkit karakter 8 jenis ( diprogram pemakai ).
4. RAM data tampilan 80 x 8 bit ( 8 karakter ).
5. Duty ratio 1/16.
6. RAM data tampilan dan RAM pembangkit karakter dapat dibaca dari unit mikroprosesor.
7. Beberapa fungsi perintah antara lain adalah penghapusan tampilan ( *display clear*), posisi kursor awal ( *crusor home*), tampilan karakter kedip ( *display character blink*), penggeseran kursor ( *crusor shift*) dan penggeseran tampilan ( *display shift*).
8. Rangkaian otomatis reset saat daya dinyalakan.
9. Catu daya tunggal +5 volt.

## 2.7 Driver Motor

Driver motor adalah rangkaian listrik yang dapat menjalankan motor dengan menggunakan sinyal kontrol sederhana. Motor membutuhkan aliran arus yang tinggi

atau tegangan yang tinggi. Maka dari itu dibutuhkan rangkaian listrik *driver* motor yang merupakan jembatan antara mikrokontroler dan motor. Dalam Gambar 2.16 ditampilkan gambar rangkaian driver motor.



Gambar 2.16 Rangkaian Driver Motor

Sumber: Irwansetyo, 2010

## 2.8 Sensor Rotary Encoder

*Rotary encoder* atau yang dikenal dengan *shaft encoder* adalah perangkat elektronika yang digunakan untuk mengkonversi sudut dari perputaran poros atau roda ke dalam kode digital. Komponen ini biasa digunakan dalam bidang robotika, perangkat komputer dan perangkat elektronik lainnya. *Rotary encoder* dibedakan menjadi dua jenis yakni *rotary encoder* absolut dan relatif.

*Rotary encoder* absolut merupakan jenis sensor yang mampu menghasilkan kode digital yang unik untuk masing-masing sudut poros. Pada prinsipnya sensor ini tersusun atas plat baja dan kontak. Plat baja dipotong dan disusun dengan pola tertentu kemudian ditempelkan pada suatu poros. Plat baja dan kontak ini secara prinsip kerja menyerupai saklar dalam kondisi ON apabila keduanya saling bersentuhan dan OFF apabila terpisah. Plat baja dan kontak diatur sedemikian rupa sehingga menghasilkan kondisi yang berbeda untuk tiap-tiap sudut poros. Bentuk fisik dari *rotary encoder* absolut ditunjukkan dalam Gambar 2.17.



**Gambar 2.17** Rotary Encoder Absolut

Sumber : Rosyidi, 2011

*Rotary encoder* relatif tidak dapat mengukur posisi sudut poros melainkan hanya mengukur perubahan sudut poros terhadap posisi sudut sebelumnya. Rotary ini digunakan ketika metode *rotary* absolut tidak dapat digunakan. Secara prinsip, sistem ini terdiri dari piringan yang dipasang pada poros dan sensor optik. Sistem ini menggunakan metode saklar optik, misal *photodiode*, untuk menghasilkan pulsa listrik yang digunakan sebagai masukan bagi rangkaian kontrol elektronika. Dalam Gambar 2.18 menunjukkan bentuk fisik *rotary encoder* relatif.



**Gambar 2.18** Rotary Encoder Relatif

Sumber: Rosyidi, 2011

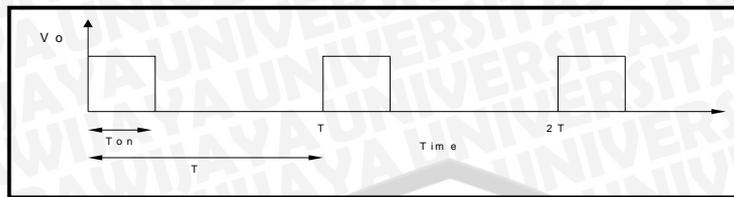
Salah satu kekurangan *Rotary encoder* adalah tidak dapat menentukan arah putaran poros. Agar dapat mengetahui arah putaran poros, pada sistem harus ditambahkan dua buah sensor optik yang dipasang pada sudut berbeda. Pembacaan arah putaran dan jumlah putaran dilakukan dengan membaca dan membandingkan sinyal keluaran dari kedua buah sensor. Tipe *rotary encoder* ini dikenal dengan *quadrature encoder*.

## 2.9 Pulse Width Modulation (PWM)

*Pulse Width Modulation* (PWM) 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 ATMEGA 32 *duty cycle* ditentukan dari *Output Compare Register 0 (OCR0)*. Sinyal PWM secara umum dapat dilihat dalam Gambar 2.19.



**Gambar 2.19** Sinyal PWM Secara Umum

Sumber: Irwansetyo, 2010

$$Duty\ cycle = \frac{T_{on}}{T} \times 100\% \quad (2.1)$$

$$V_{DC} = Duty\ cycle \times V_{CC} \quad (2.2)$$

Sedangkan frekuensi sinyal dapat ditentukan dengan rumus sebagai berikut:

$$f_{OCn} = \frac{f_{clk\_I/O}}{N \cdot 256} \quad (2.3)$$

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

## 2.10 Buzzer Alarm

Buzzer adalah sebuah komponen elektronika yang berfungsi untuk mengubah getaran listrik menjadi getaran suara. Pada dasarnya prinsip kerja buzzer hampir sama dengan loud speaker, jadi buzzer juga terdiri dari kumparan yang terpasang pada diafragma dan kemudian kumparan tersebut dialiri arus sehingga menjadi elektromagnet, kumparan tadi akan tertarik ke dalam atau keluar, tergantung dari arah arus dan polaritas magnetnya, karena kumparan dipasang pada diafragma maka setiap gerakan kumparan akan menggerakkan diafragma secara bolak-balik sehingga membuat udara bergetar yang akan menghasilkan suara. Buzzer biasa digunakan sebagai indikator bahwa proses telah selesai atau terjadi suatu kesalahan pada sebuah alat (alarm).

Dalam Gambar 2.20 menunjukkan bentuk fisik *Buzzer Alarm*



Gambar 2.20 Buzzer Alarm



## BAB III

### METODOLOGI PENELITIAN

#### 3.1 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 dari mikrokontroler dan rangkaian sensor.
5. Perancangan perangkat lunak *software* pengontrolan.

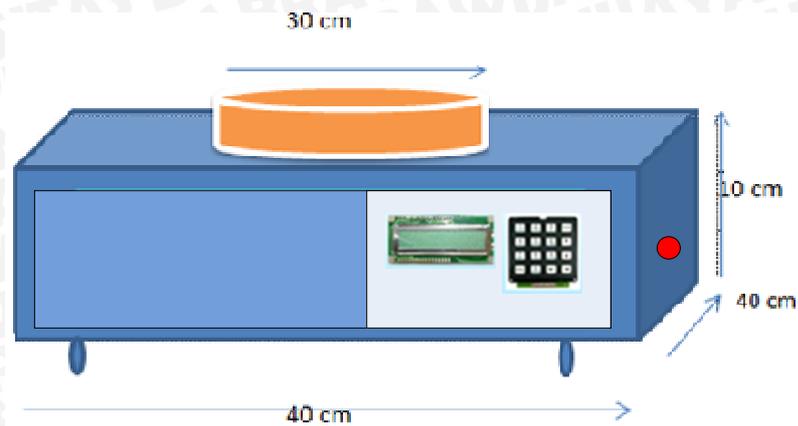
#### 3.2 Pembuatan Alat

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

##### 3.2.1 Pembuatan Pemutar Gerabah

Pembuatan pemutar gerabah dilakukan dengan pembuatan desain pemutar gerabah terlebih dahulu kemudian dilanjutkan dengan pembuatan kerangka pemutar gerabah dan penyusunan motor arus searah atau *direct current* (DC) pada pemutar gerabah.

Pembuatan alat dengan menggunakan komponen elektronika yang telah direncanakan. Pembuatan alat untuk perangkat keras juga meliputi penambahan rangkaian sensor, LCD, keypad, dan perkabelan. Dalam Gambar 3.1 akan ditampilkan gambar rancangan perangkat keras dari pemutar gerabah.



Gambar 3.1 Desain Rancangan Perangkat Keras

### 3.3 Perancangan dan Penyusunan Perangkat Lunak

Penyusunan perangkat lunak digunakan untuk mengendalikan dan mengatur kerja dari alat ini. Desain dan parameter yang telah dirancang kemudian diterapkan kedalam mikrokontroler dengan menggunakan bahasa C dan *compiler CodeVision AVR*.

### 3.4 Pengujian Alat

Pengujian alat dilakukan untuk memastikan bahwa sistem ini berjalan sesuai yang direncanakan. Pengujian alat meliputi pengujian perangkat keras yang dilakukan baik tiap blok maupun keseluruhan sistem.

#### 3.4.1 Pengujian Tiap Blok

Pengujian tiap blok dilakukan dengan tujuan untuk menyesuaikan nilai masukan dan nilai keluaran tiap-tiap blok sesuai dengan perancangan yang dilakukan sebelumnya.

#### 3.4.2 Pengujian Keseluruhan Sistem

Pengujian sistem secara keseluruhan dilakukan dengan tujuan untuk mengetahui unjuk kerja alat setelah perangkat keras dan perangkat lunak diintegrasikan bersama.

### 3.5 Pengambilan Kesimpulan dan Saran

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 penyempurnaannya. Saran

diberikan untuk memperbaiki kesalahan, dan kemungkinan pengembangan alat agar lebih baik untuk penelitian selanjutnya.



## BAB IV PERANCANGAN DAN PEMBUATAN ALAT

Perancangan dan pembuatan alat ini terdiri dari dua bagian, yaitu perancangan dan pembuatan perangkat keras serta perancangan dan pembuatan perangkat lunak. Perancangan dan pembuatan alat dilakukan secara bertahap untuk memudahkan analisis sistem. Beberapa aspek yang perlu dijelaskan dalam bab ini meliputi penentuan spesifikasi alat, perencanaan masing-masing blok rangkaian serta perencanaan sistem secara keseluruhan.

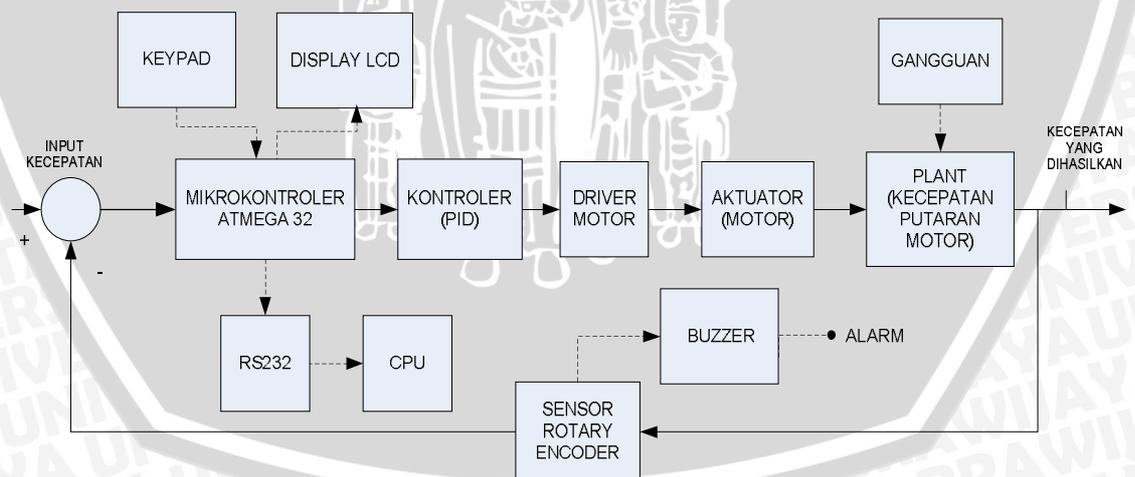
Perancangan ini terdiri dari:

- Perancangan sistem.
- Perancangan perangkat keras (alat pemutar gerabah, catu daya sistem, sensor *rotary encoder*, *driver motor*, dan mikrokontroler).
- Perancangan sistem kontroler PID.
- Perancangan perangkat lunak.

### 4.1. Perancangan Sistem

#### 4.1.1 Blok Diagram Sistem

Blok diagram sistem yang dirancang ditunjukkan dalam Gambar 4.1.



**Gambar 4.1** Gambar Blok Diagram Sistem (perancangan)

Pada sistem mekanik pemutar gerabah ini menggunakan satu motor DC sebagai penggeraknya. Motor ini dihubungkan oleh sebuah poros yang berfungsi sebagai aktuator. Karena berfungsi sebagai aktuator, maka diharapkan piringan pemutar gerabah dapat berputar seiring dengan berputarnya motor DC sebagai penggerak. Pada alat ini

dipasang sebuah sensor *rotary encoder* yang berfungsi menghitung kecepatan putaran motor. Sensor ini terhubung dengan Mikrokontroler ATMEGA 32.

Alat pemutar gerabah menggunakan penggerak motor (*driver motor*) menggunakan system *Pulse Width Modulation* (PWM). *Input setting point* dilakukan dengan menekan tombol keypad yang berupa data digital, yang kemudian dikonversi ke tegangan oleh mikrokontroler ATMEGA 32. Keluaran dari mikrokontroler berupa PWM menjadi masukan untuk Penggerak motor (*driver motor*), di mana kecepatan keluaran dari motor akan dideteksi oleh sensor *rotary encoder*.

Mikrokontroler ATMEGA 32 berfungsi sebagai mikrokontroler yang mengolah seluruh data masukan dari sensor dan *driver* pengendali motor, selain itu juga mengolah seluruh perhitungan kontroler PID. Kemudian data tersebut dibandingkan dengan *set point* untuk kemudian dicari nilai *error*nya dan dilakukan penghitungan untuk nilai kontroler yang diperlukan. Mikrokontroler ini berhubungan dengan *driver* motor yang berfungsi sebagai pengkondisi sinyal dari sinyal kontrol mikrokontroller menjadi sinyal untuk menggerakkan aktuatornya yaitu motor DC. *Driver* motor ini berfungsi mengeksekusi hasil perhitungan dari mikrokontroler. Mikrokontroler ATMEGA 32 juga terhubung dengan *buzzer* yang dilengkapi dengan *alarm* yang berfungsi untuk memberi peringatan apabila motor berputar lambat terlalu jauh di bawah *setpoint* karena beban yang terlalu berat.

## 4.2. Perancangan Perangkat Keras (*hardware*)

Berdasarkan diagram blok pada Gambar 4.1 perancangan alat yang telah disusun, perancangan perangkat keras terdiri dari alat pemutar gerabah, sensor *rotary encoder*, rangkaian driver motor, motor DC, Mikrokontroler.

### 4.2.1. Spesifikasi Alat

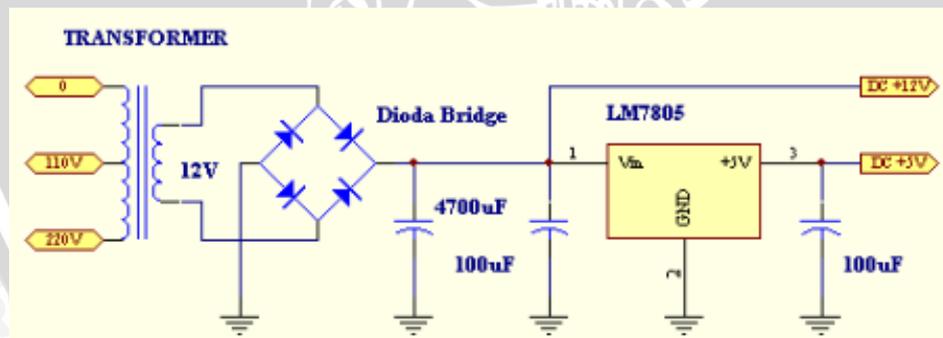
Rangka mekanik berbahan dasar kayu dengan panjang 40 cm dan lebar 40 cm. Motor penggerak alat putar berjumlah 1 buah dan dihubungkan langsung dengan bagian poros aluminium yang berfungsi memutar piringan pemutar yang berdiameter 30 cm. Sensor *rotary encoder* dipasang pada poros yang terhubung motor. Sensor terpasang lurus langsung dengan motor, agar kecepatan motor langsung terdeteksi oleh sensor. Bentuk mekanik alat pemutar gerabah dapat dilihat dalam Gambar 4.2.



Gambar 4.2 Gambar Rancangan Mekanik

#### 4.2.2. Rangkaian Catu Daya Sistem

Sistem mikrokontroler membutuhkan catu daya sebesar 5 volt agar dapat bekerja. Catu daya 5 volt sudah memenuhi tegangan kerja mikrokontroler ATMEGA 32 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. Dalam Gambar 4.3 berikut akan ditampilkan rangkaian catu dayanya.



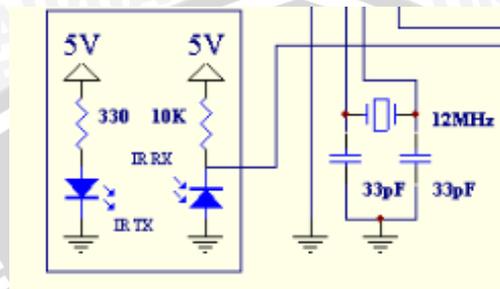
Gambar 4.3 Rangkaian Catu Daya Sistem (perancangan)

Tegangan sumber yang digunakan dalam rangkaian catu daya adalah 24 volt. Nilai keluaran regulator langsung berhubungan dengan mikrokontroler, rangkaian sensor, dan juga rangkaian *driver*.

#### 4.2.3. Sensor Rotary Encoder

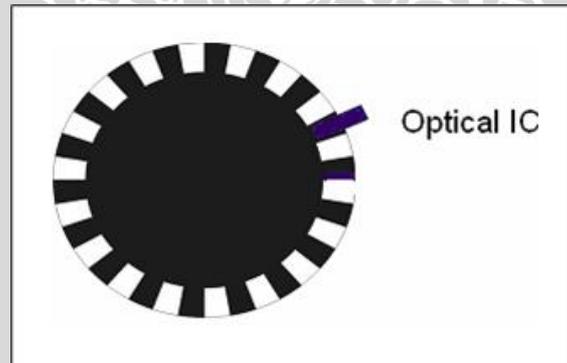
Sensor putaran (*rotary encoder*) yang digunakan adalah sensor putaran berbasis optik menggunakan komponen *Optical IC* dan piringan berpola. Sensor

rotari diletakkan pada sebuah roda yang dihubungkan dengan sebuah pegas dan tidak dipengaruhi oleh roda pergerakan. Komponen ini merupakan kombinasi LED inframerah sebagai pemancar cahaya, regulator tegangan, penguat, dan *phototransistor* NPN sebagai penerima cahaya. Dalam Gambar 4.4 menunjukkan rangkaian sensor *rotary encoder*.



**Gambar 4.4** Rangkaian Sensor *Rotary Encoder*(perancangan)

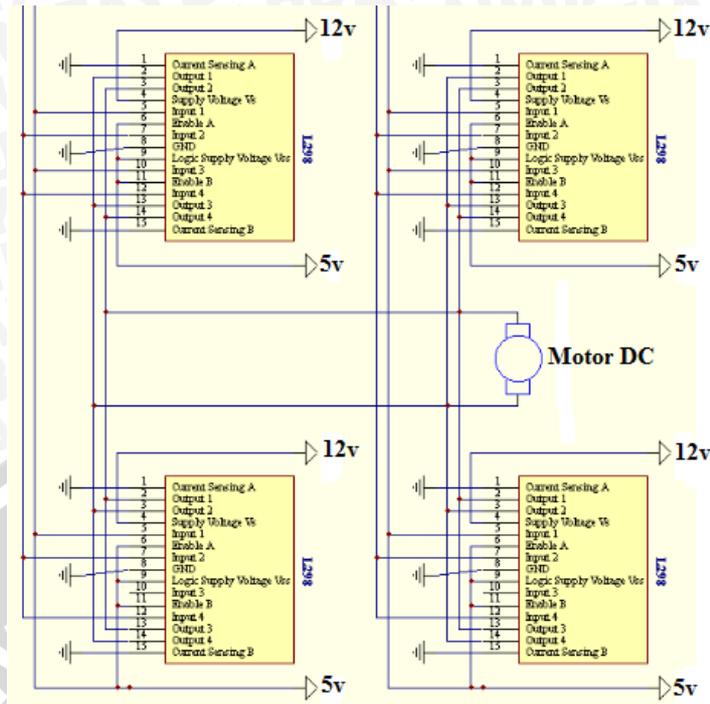
Resolusi sensor yang digunakan pada perancangan ini sebesar 60 pulsa untuk setiap putaran penuh. Sifat sinyal *incremental* didapatkan dengan membuat pola berlubang pada sisi luar piringan berpola seperti ditunjukkan dalam Gambar 4.5.



**Gambar 4.5** Letak *Optical IC* Pada Piringan Berpola

#### 4.2.4. Perancangan *Driver Motor*

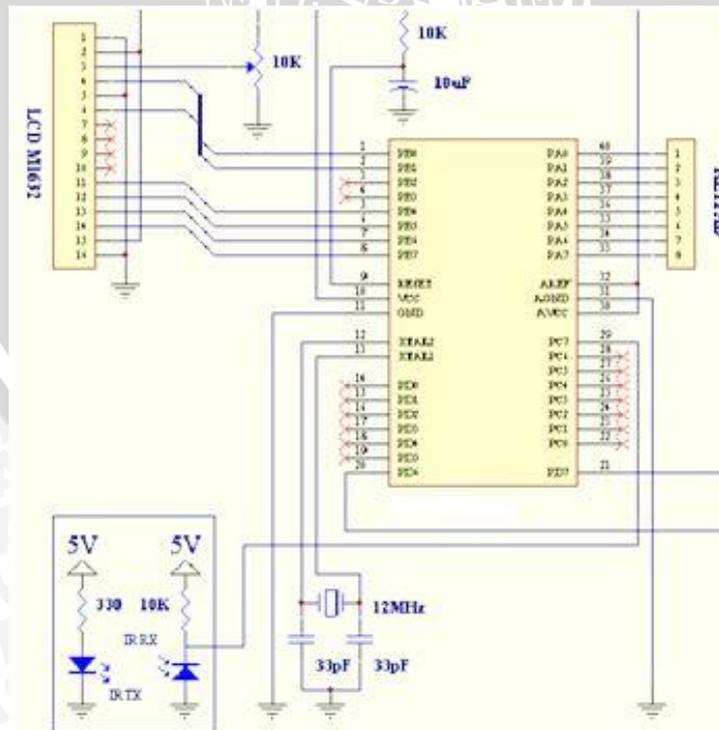
Driver pada rangkaian ini menggunakan *riley* untuk logika *ON-OFF* nya dan untuk mengontrol putaran motor digunakan E-MOSFET kanal N dengan masukan berupa sinyal PWM dari mikrokontroler. Rangkaian *driver* motor akan ditunjukkan dalam Gambar 4.6.



Gambar 4.6 Rangkaian Driver Motor (perancangan)

#### 4.2.5 Rangkaian Mikrokontroler

Pada alat pemutar gerabah ini digunakan mikrokontroler ATMEGA 32 sebagai pengolah utama sebagai pengolah data dari sensor *rotary encoder*, pengendali arah dan kecepatan motor serta melakukan proses PID. Konfigurasi kaki I/O dari mikrokontroler ATMEGA 32 ditunjukkan dalam Gambar 4.7.



**Gambar 4.7** Konfigurasi Kaki I/O dari Mikrokontroler ATMEGA 32( Perancangan)

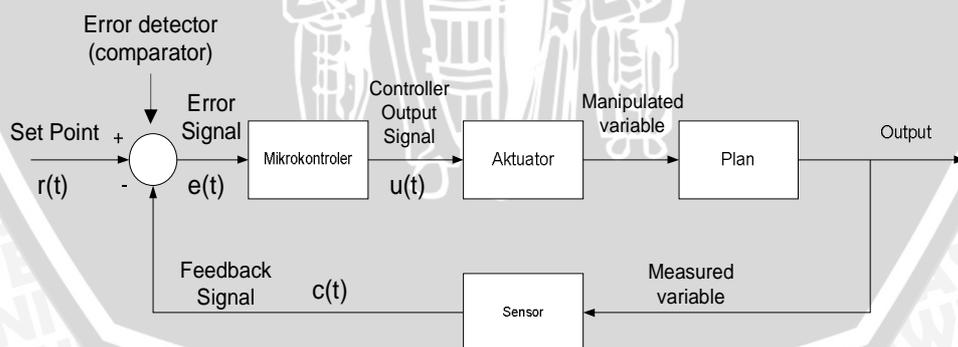
Mikrokontroler ATMEGA 32 memiliki kaki sebanyak 40 pin, 32 diantaranya dapat diprogram menjadi masukan atau keluaran. Berikut merupakan tabel fungsi pin mikrokontroler pada Tabel 4.1

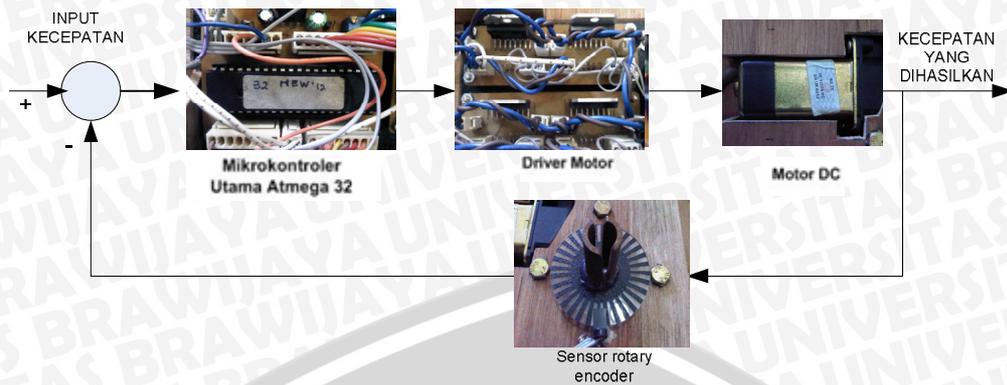
**Tabel 4.1** Fungsi Pin Mikrokontroler

No	Pin	Fungsi
1	PA0- PA7	Jalur masukan untuk <i>keypad</i>
2	PB0	Jalur keluaran LCD
3	PB1	Jalur keluaran LCD
4	PB4	Jalur keluaran LCD
5	PB5	Jalur keluaran LCD
6	PB6	Jalur keluaran LCD
7	PB7	Jalur keluaran LCD
8	PC7	Jalur masukan <i>Rotary Encoder</i>
9	PD6	Jalur keluaran motor
10	PD7	Jalur keluaran motor

### 4.3 Perancangan parameter *PID*

Berikut ini adalah blok aplikasi sistem kontroler PID pada alat pemutar gerabah ditunjukkan dalam Gambar 4.8



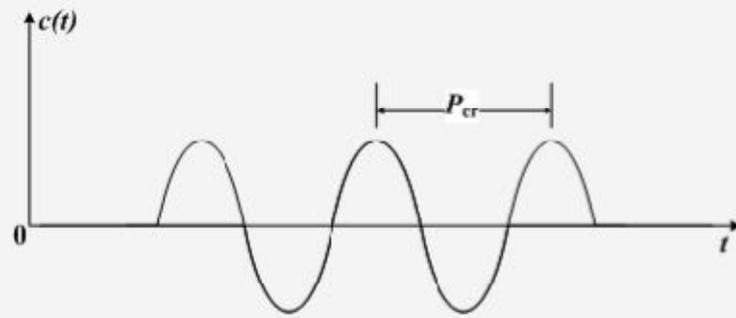


**Gambar 4.8** Diagram Blok Aplikasi Perancangan Kontroler PID pada alat pemutar gerabah.

Pada proses *tuning* kontrol PID dengan menggunakan metode osilasi Ziegler-Nichols dimulai dengan memberikan nilai  $\infty$  pada parameter  $T_i$  dan 0 pada parameter  $T_d$ . Sedangkan nilai  $K_p$  dinaikkan sedikit demi sedikit hingga didapatkan grafik yang berkesinambungan. Kesinambungan yang dimaksud adalah saat grafik memiliki amplitudo yang sama pada setiap periodenya. Setelah didapatkan grafik yang berkesinambungan langkah selanjutnya adalah menghitung nilai  $K_{cr}$  dan  $P_{cr}$ .  $K_{cr}$  adalah nilai  $K_p$  saat terjadi osilasi berkesinambungan sedangkan  $P_{cr}$  adalah periode kesinambungan dari grafik. Setelah didapatkan  $K_{cr}$  dan  $P_{cr}$  langkah selanjutnya adalah menghitung nilai  $K_p$ ,  $T_i$  dan  $T_d$  sesuai dengan aturan dari Ziegler-Nichols.

#### 4.3.1 Tuning Eksperimen

*Tuning* eksperimen adalah proses yang dilakukan untuk mendapatkan hasil kontroler yang optimal dengan cara suatu percobaan. Inti dari *tuning* eksperimen adalah menentukan nilai dari tiga buah parameter yang terdapat pada kontroler PID yaitu konstanta proporsional ( $K_p$ ), konstanta integral ( $K_i$ ) dan konstanta diferensial ( $K_d$ ). Pada perancangan kontroler PID pemutar gerabah ini, menggunakan *tuning* parameter Ziegler-Nichols metode ke dua yang bertujuan untuk mencapai 25% *maximum overshoot* pada respon unit step. Metode ini berguna untuk sistem yang mungkin mempunyai *step response* berosilasi terus menerus dengan teratur seperti pada Gambar 4.9



**Gambar 4.9.** Osilasi Berkesinambungan dengan Periode  $P_{cr}$

Sumber: Ogata, K., 1997

Sehingga metode ke dua Ziegler-Nichols ini dipilih karena keluaran sistem memiliki osilasi berkesinambungan untuk nilai  $K_p$  manapun yang telah diambil dari sensor *rotary encoder* yang memiliki osilasi terhadap pergerakan motor. Langkah atau acuan penentuan parameter  $K_p$ ,  $K_i$  dan  $K_d$  didasarkan pada aturan Ziegler-Nichols metode ke dua. Metode dilakukan dengan eksperimen dengan memberikan kontroler P pada suatu sistem *close loop* dengan *plant* terpasang yang ditunjukkan dalam perancangan Gambar 4.14.



**Gambar 4.10.** Perancangan Sistem Ziegler-Nichols Metode ke Dua

Sumber: Ogata, K., 1997

Langkah metode tersebut ialah sebagai berikut:

- 1) Buat suatu sistem loop tertutup dengan kontroler P dan *plant* di dalamnya
- 2) Mula-mula yang dilakukan adalah membuat  $T_i = \infty$  dan  $T_d = 0$ .
- 3) Kemudian hanya dengan menggunakan tindakan kontrol proporsional, harga ditingkatkan dari nol kesuatu nilai kritis  $K_{cr}$ , disini mula-mula keluaran memiliki osilasi yang berkesinambungan.
- 4) Dari keluaran yang berosilasi secara berkesinambungan, penguatan kritis  $K_{cr}$  dan periode  $P_{cr}$  dapat ditentukan.
- 5) Nilai *sampling time* (waktu cuplik) juga mempengaruhi perhitungan PID, tentunya saat penggunaan kontrol integral dan diferensial.
- 6) Menghitung nilai  $K_p$ ,  $T_i$  dan  $T_d$  sesuai dengan aturan dari Ziegler-Nichols yaitu,

$$K_p = 0.6 \times K_{cr}$$

$$T_i = 0.5 \times P_{cr}$$

$$Td = 0.125 \times Pcr$$

Nilai Ki dan Kd didapatkan dengan menggunakan perhitungan sebagai berikut,

$$Ki = \frac{Kp}{Ti}$$

$$Kd = Kp \times Td$$

#### 4.4 Perancangan Perangkat Lunak

##### 4.4.1 Perancangan Algoritma Utama Kontroler PID

Pembuatan program kontroler PID ini dilakukan berdasarkan persamaan kontroler PID digital. Kontroler PID digital merupakan bentuk lain dari kontroler PID yang diprogram dan dijalankan menggunakan komputer atau mikrokontroler. Untuk dapat mengimplementasikan PID digital di komputer atau mikrokontroler, maka kontroler PID analog harus diubah terlebih dahulu ke bentuk digital.

Penurunan kontroler PID digital dapat dilihat pada Persamaan 4.1 sampai 4.5.

$$u(t) = K_p e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \dots\dots\dots(4-1)$$

diamana  $\frac{1}{T_i} = Ki$  dan  $T_d = Kd$

Bentuk integral dan differensial dapat ditulis dalam bentuk diskrit seperti pada persamaan

$$\int_0^t e(t) dt \approx T \sum_0^k e(k) \dots\dots\dots(4-2)$$

$$\frac{de(t)}{dt} \approx \frac{e_k - e_{k-1}}{T} \dots\dots\dots(4-3)$$

$$u(t) = K_p e(k) + K_i T \sum_0^k e(k) + \frac{1}{T} K_d (e_k - e_{k-1}) \dots\dots\dots(4-4)$$

$$u = K_p \text{error} + K_i (\text{error} * dt + \text{last\_error})xTs + \frac{K_d}{Ts} (\text{error} - \text{last\_error}) \dots\dots\dots(4-5)$$

Dimana:

u(t)= sinyal *output* pengendali PID

Kp = konstanta proporsional

Ti = waktu integral

Td = waktu derivatif

Ki = konstanta integral (Kp/Ti)



$K_d$  = konstanta derivatif ( $K_p.T_d$ )

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

*error* ialah nilai kesalahan

*last\_error* ialah nilai kesalahan sebelumnya

$T_s$  ialah *sampling time* (waktu cuplik)

Pembuatan program kontroler PID ini dilakukan berdasarkan persamaan kontroler PID digital pada persamaan (4-4). Pertama-tama akan dibuat dua buah variabel berupa *error*, *last\_error*. Gunanya adalah untuk menyimpan data *error* dan *last\_error* yang akan digunakan pada perhitungan aksi kontroler PID. Setiap satu kali *looping* program, *error* akan diperbaharui dengan data yang diambil dari sensor, dan sebelumnya akan disimpan di *last\_error*. Keluaran dari perhitungan program kontroler PID ini adalah nilai PWM.

Keluaran kontroler berupa nilai PWM ini akan memanggil fungsi pengendali *driver* motor. Nilai PWM keluaran dari kontroler PID ini akan ditambahkan dan dikurangkan terlebih dahulu dengan *base* PWM untuk motor dan dijumlahkan dengan *offset* PWM pada motor.

Nilai-nilai variabel ditunjukkan pada persamaan 4.6 sampai 4.8.

$$\text{Error} = \text{SP-PV} \dots\dots\dots(4-6)$$

$$u(t) = K_p e(k) + K_i T \sum_0^k e(k) + \frac{1}{T} K_d (e_k - e_{k-1}) \dots\dots\dots(4-7)$$

$$\text{PWM motor} = \text{Base PWM} + \text{PID} \dots\dots (4-8)$$

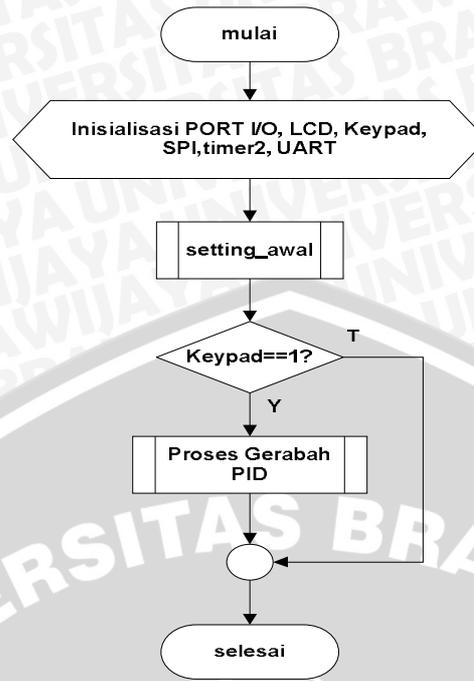
Deviasi/simpangan antar variabel terukur (PV) dengan nilai acuan (SP) disebut *error*. *Setpoint* (SP) adalah suatu parameter nilai acuan atau nilai yang diinginkan. *Present Value* (PV) adalah nilai pembacaan sensor saat itu atau variabel terukur yang di umpan balik oleh sensor (sinyal *feedback*).

#### 4.4.2 Perancangan Diagram Alir Program Kontroler PID

Diagram alir program utama kontroler PID meliputi proses inialisasi, *tuning* parameter, akumulasi error dan perhitungan PID sebagai proses berjalannya motor.

Diagram alir program utama ditunjukkan dalam Gambar 4.3.





**Gambar4.11** Diagram Alir Program Mikrokontroler

Program utama mikrokontroler berisi inisialisasi PORT I/O, LCD, Keypad, komunikasi SPI, timer2 untuk motor dan UART. Selain itu juga terdapat inisialisasi variabel-variabel yang akan digunakan oleh program.

#### 4.4.3 Diagram Alir Subrutin Proses PID Sensor

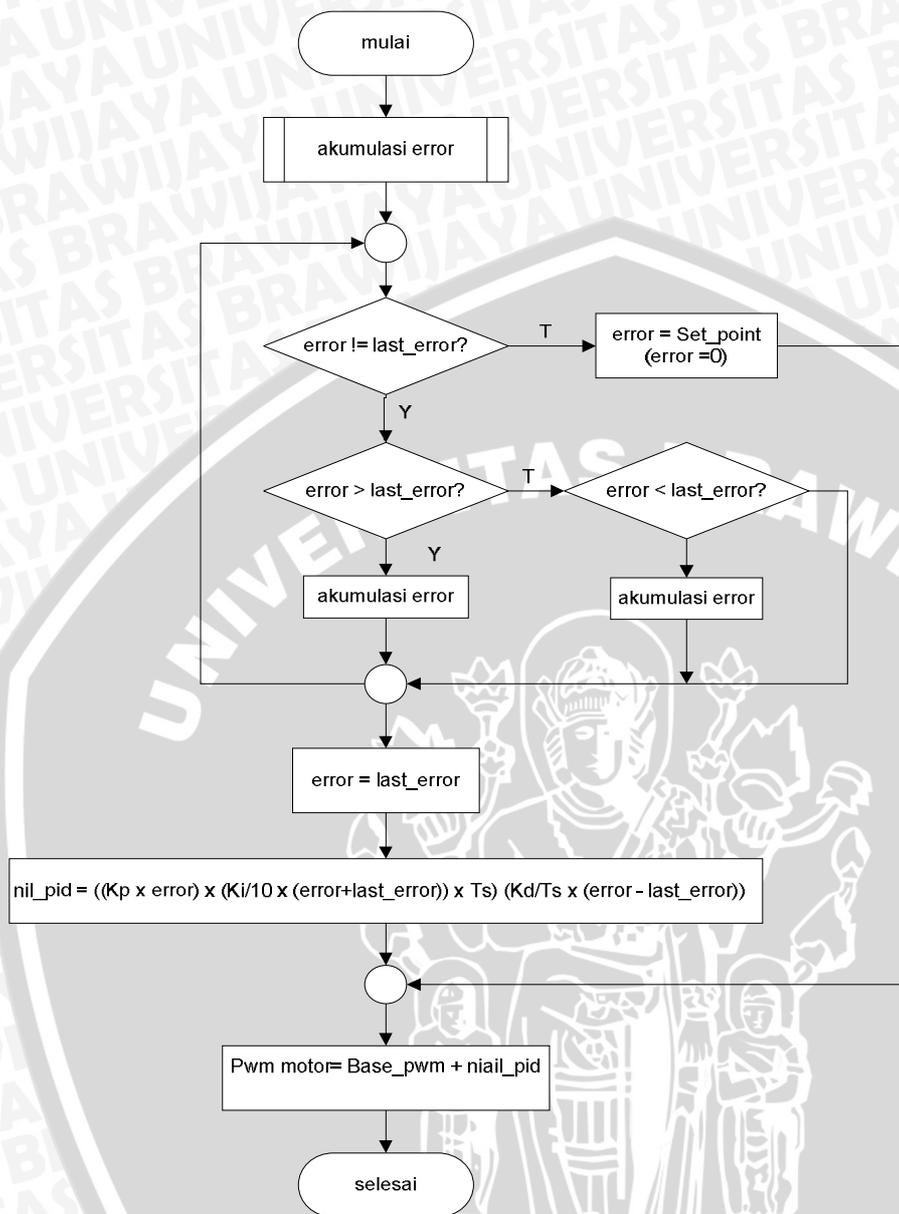
Proses PID Sensor dalam pemutar gerabah ini yaitu sensor *rotary encoder*. Data Nilai pembacaan sensor *rotary encoder* diolah dengan menggunakan kontroler PID sesuai dengan blok diagram kontrol PID. Diagram alir subrutin proses PID sensor

ditunjukkan

dalam

Gambar

4.12.



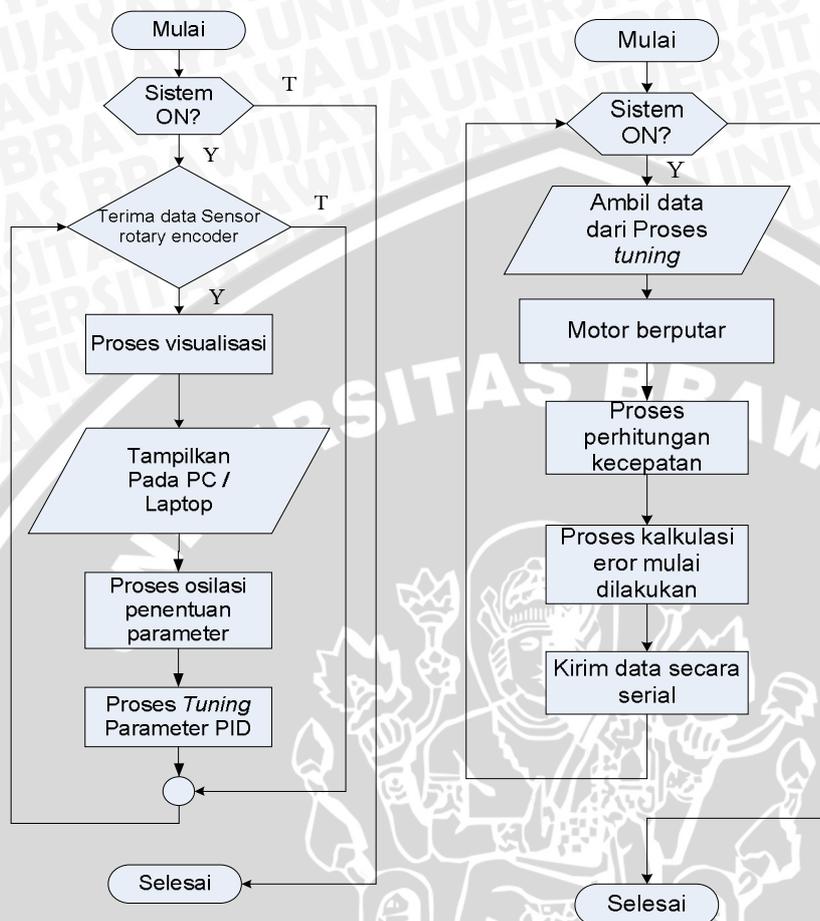
Gambar 4.12 Diagram Alir Subrutin Proses PID Sensor

Subrutin hitung PID ini berisi *listing* program untuk proses perhitungan kontroler PID dimana terdapat *loopingerror* secara terus menerus untuk mendapatkan akumulasi *error* yang akurat.

#### 4.4.4 Diagram Alir Sistem Keseluruhan

Diagram alir sistem keseluruhan ini meliputi diagram alir proses tuning parameter dimana saat pengambilan data  $K_p$ ,  $K_i$  dan  $K_d$  yang didasarkan pada aturan Ziegler-Nichols metode kedua dan diagram alir proses berjalanya pemutar gerabah

setelah proses tuning selesai dilakukan. Diagram alir sistem keseluruhan ditunjukkan dalam Gambar 4.13.



(a)

(b)

**Gambar 4.13** (a) Diagram Alir Proses Tuning Parameter pada Perangkat Penerima, (b) Diagram Alir Proses berputarnya alat gerabah.

## BAB V

### PENGUJIAN DAN ANALISIS SISTEM

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 sensor *rotary encoder*
- Pengujian motor dan *driver* motor
- Pengujian keseluruhan sistem
  - Pengujian pengontrolan mesin pemutar gerabah tanpa beban
  - Pengujian pengontrolan mesin pemutar gerabah dengan beban yang berbeda- beda.

#### 5.1 Pengujian Sensor *Rotary Encoder*

Pengujian ini bertujuan untuk mengetahui level tegangan sinyal listrik oleh rangkaian *optocoupler* dan untuk mengetahui keberhasilan rangkaian sensor *rotary encoder* dalam mendeteksi putaran. Pengujian respon rangkaian sensor *rotary encoder* menggunakan *oscilloscope* untuk melihat karakteristik perubahan tingkat logika dari rangkaian sensor. Motor arus searah atau *direct current* (DC) terhubung dengan piringan berlubang (32 lubang) diberi sumber 12 V agar berputar.

Berdasarkan *datasheet*, *Optical IC GP1A25LC* produksi *SHARP Corporation* masih dapat membaca putaran dengan frekuensi maksimum 16000 Hz. Dengan menggunakan nilai frekuensi maksimum tersebut dapat dihitung kecepatan putaran motor maksimum yang dapat dideteksi oleh rangkaian sensor rotari menggunakan persamaan berikut.

$$\omega_{maks} = \frac{F_{maks}}{32} \quad (5.1)$$

$$\omega_{maks} = \frac{16000 \text{ Hz}}{32}$$

$$\omega_{maks} = 500 \text{ rpm}$$

Berdasarkan persamaan 5.1 diketahui bahwa kecepatan putaran motor maksimum yang dapat dideteksi oleh sensor rotari adalah 500 rpm.

### 5.1.1 Peralatan Pengujian

Dalam pengujian sensor *rotary encoder* ini digunakan beberapa peralatan, diantaranya:

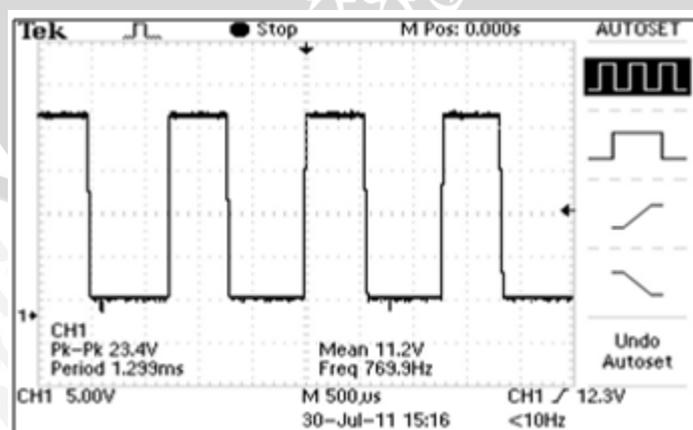
1. Motor arus searah atau *direct current* (DC)
2. Osiloskop dan *probe* osiloskop
3. Rangkaian pengkondisi sinyal *Rotary Encoder*
4. Sensor *Rotary Encoder*

### 5.1.2 Prosedur Pengujian

Prosedur pengujian dilakukan dengan menghubungkan *probe oscilloscope* pada rangkaian pengkondisi sinyal rotari. *Channel 1* pada osiloskop dihubungkan dengan pin *output* LM741 dan *channel 2* pada osiloskop dihubungkan dengan *ground* rangkaian. Selanjutnya motor DC yang terhubung dengan sensor *Rotary Encoder* diberi tegangan maksimal agar berputar.

### 5.1.3 Hasil Pengujian

Setelah melakukan prosedur pengujian, maka pada osiloskop akan menunjukkan tampilan seperti dalam Gambar 5.1.



Gambar 5.1 Hasil pengujian *Rotary Encoder* menggunakan osiloskop

Berdasarkan Gambar 5.1 didapatkan *rise time* dan *fall time* pulsa saat kecepatan motor maksimal:

*Peak to peak* (Tegangan puncak ke puncak) = 23,4V

*Period* (Waktu yang diperlukan membentuk satu gelombang penuh) = 1,299ms

*Frequency* (Banyak gelombang yang terbentuk dalam satuan waktu) = 769,9Hz

Berikut akan ditampilkan pula hasil pengujian sensor *rotary encoder* dengan nilai

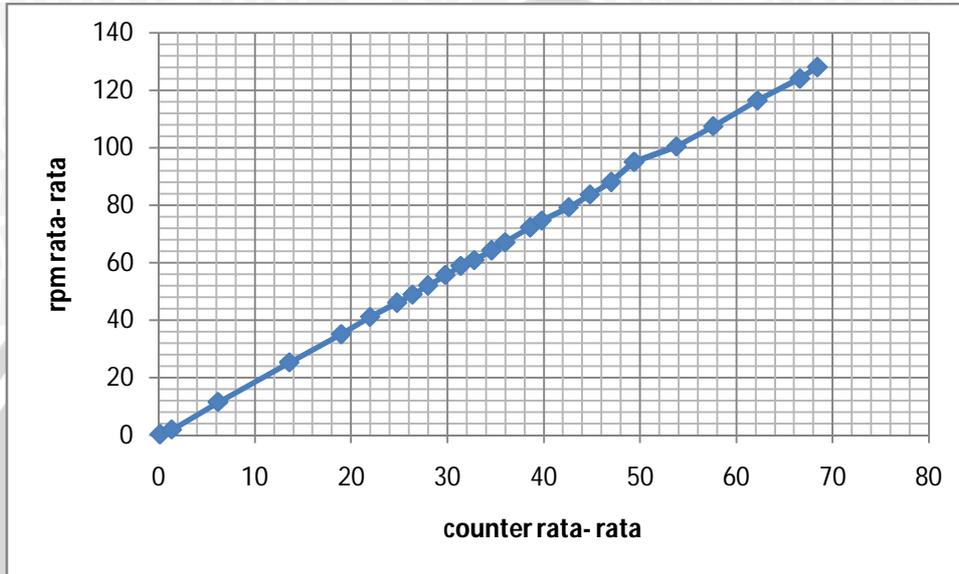
*Counter* dan rpm (*radiant per minute*) yang ditunjukkan pada Tabel 5.1

**Tabel 5.1** Hasil Pengujian *Counter* dan Rpm

rpm rata- rata	<i>counter</i> rata- rata
0,2	0,2
1,8	1,4
11,4	6,2
25,2	13,6
35	19
41	22
46	24,8
48,8	26,4
52	28
55,6	29,8
58,8	31,4
60,8	32,8
64,2	34,6
67	36
72,2	38,6
74,6	39,8
79,2	42,6
83,6	44,8
88	47
95	49,4
100,4	53,8

107,4	57,6
116,4	62,2
124	66,6
128	68,4

Berikut merupakan grafik dari pengujian *counter* dan rpm pada Gambar 5.2



Gambar 5.2 Grafik Hubungan Antara Rpm dan Counter

## 5.2 Pengujian Motor dan Driver Motor

Tujuan dari pengujian ini adalah untuk mengetahui *output* dari *driver* motor apabila diberi *input* yang berbeda-beda.

### 5.2.1 Peralatan yang digunakan

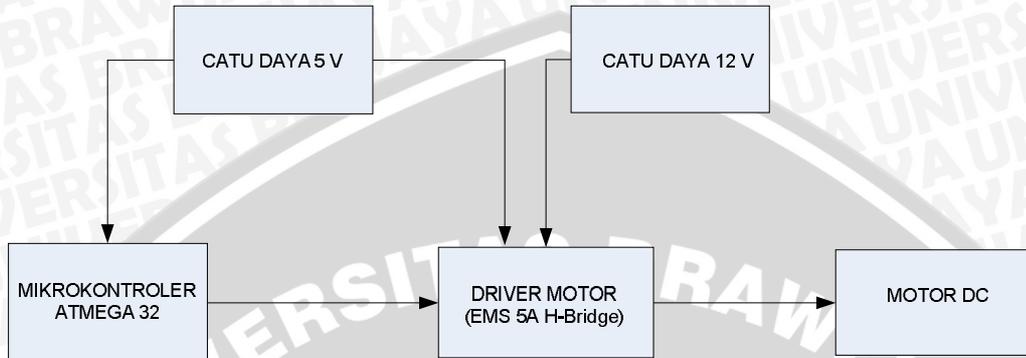
Dalam pengujian sensor *rotary encoder* ini digunakan beberapa peralatan, diantaranya:

1. Minimum sistem mikrokontroler ATmega 32
2. *Driver* motor
3. Motor DC
4. Catu daya 5V dan 12V

### 5.2.2 Prosedur pengujian

1. Merangkai peralatan seperti pada gambar 5.3
2. Mengaktifkan catu daya 5 volt dan 12 volt
3. Menghitung besar PWM dan tegangan dari mikrokontroler

4. Menghitung besar tegangan keluaran dari *driver* motor yang diperlukan untuk motor
5. Mencatat pergerakan motor DC
6. Mencatat nilai Rpm yang dihasilkan oleh motor



Gambar 5.3 Diagram Blok Pengujian *Driver* Motor DC

### 5.2.3 Hasil Pengujian

Hasil pengujian kecepatan motor dengan nilai *Pulse Width Modulation* (PWM) yang ditentukan ditunjukkan pada tabel 5.2.

Tabel 5.2 Hasil Pengujian *Pulse Width Modulation* (PWM)

PWM	V MK (Volt)	V Motor (Volt)	Rpm
10	0.215	0.28	0.2
20	0.410	0.61	1.8
30	0.410	0.61	11.4
40	0.795	1.29	25.2
50	0.984	1.63	35
60	1.176	2.08	41
70	1.351	2.76	46
80	1.542	2.87	48.8
90	1.752	3.45	52
100	1.944	4.60	55.6
110	2.136	5.50	58.8
120	2.329	6.20	60.8

130	2.522	6.70	64.2
140	2.715	7.40	67
150	2.907	7.77	72.2
160	3.102	8.20	74.6
170	3.292	8.54	79.2
180	3.488	8.94	83.6
190	3.660	9.20	88
200	3.870	9.53	90.8
210	4.050	9.90	95
220	4.250	9.95	100.4
230	4.420	10.10	107.4
240	4.600	10.38	116.4
250	4.780	10.54	124
255	4.900	10.67	128

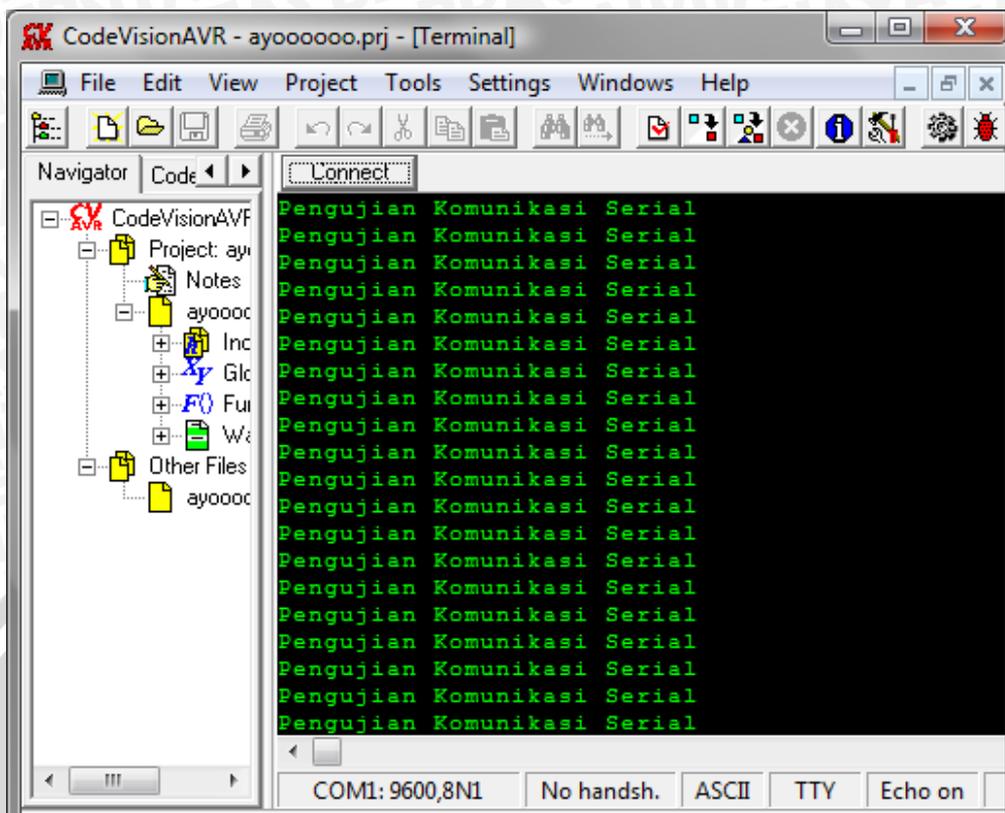
### 5.3 Pengujian Komunikasi Serial UART ke PC

Pengujian ini dilakukan untuk mengetahui apakah rangkaian mikrokontroler dapat mengirimkan data melalui komunikasi serial UART. Untuk pengujian ini, mikrokontroler mengirimkan data pada terminal komputer menggunakan konfigurasi *baudrate* 9600 bps, 8 bit data, tanpa paritas dan 1 stop bit. Data yang dikirimkan merupakan paket data yang berisi pembacaan program. Prosedur pengujian dilakukan dengan menghubungkan mikrokontroler ATmega32, kabel serial RS-232, dan Komputer sesuai Gambar 5.4



**Gambar 5.4** Diagram Pengujian Komunikasi serial UART RS232

Mikrokontroler ATmega32 akan memberikan instruksi sinyal pada pin Tx dan Rx yang terhubung oleh kabel serial RS-232 yang kemudian tersambung juga oleh komputer melalui USB. Pin instruksi Data yang dikirimkan merupakan paket data yang berisi pembacaan program. Hasil pengujian ditunjukkan pada Gambar 5.5



**Gambar 5.5** Hasil pengujian komunikasi serial dengan RS-232

Berdasarkan hasil pengujian yang telah dilakukan, dapat disimpulkan bahwa data dapat dikirim dengan baik tanpa ada karakter yang hilang dan program pembacaan pada mikrokontroler berjalan dengan benar sesuai perancangan dan perangkat USART melalui kabel serial yang terhubung dengan komputer.

## 5.4 Pengujian Sistem Secara Keseluruhan

Pengujian ini bertujuan untuk mengetahui kemampuan kerja dari perangkat lunak PID dan perangkat keras saat diintegrasikan secara bersama-sama. Pengujian dilakukan dengan merangkai rangkaian kemudian mengaktifkan semua alat yang telah dirakit, setelah itu mengamati kinerja dari sistem tersebut.

### 5.4.1 Peralatan Pengujian

1. Sensor *Rotary Encoder*
2. Beban 1 kg, 2 kg, dan 3 kg.
3. Mikrokontroler ATMEGA 32
4. Rangkaian Driver Motor

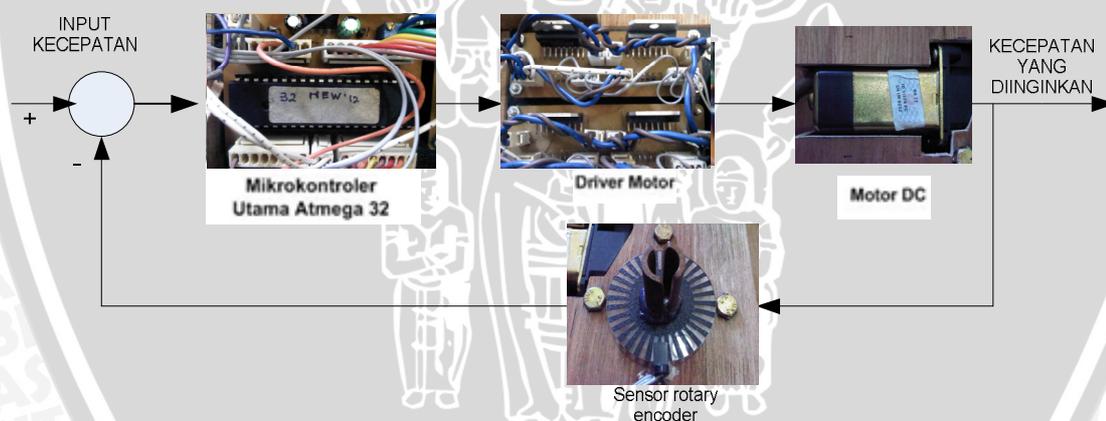
5. Motor DC
6. LCD
7. UART RS232
8. Perangkat *Writer*
9. Software *CodeVision AVR*
10. Software *MyWrite*

#### 5.4.2 Prosedur Pengujian

Prosedur pengujiannya adalah:

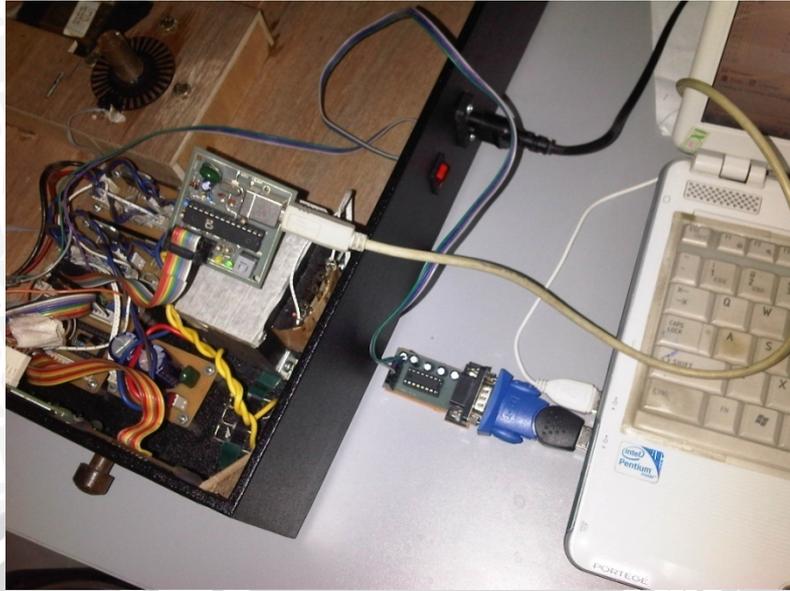
1. Membuat program untuk pengaturan motor gerabah
2. Mengatur nilai set point pada mikrokontroller
3. Melakukan tuning PID
4. Memasukkan nilai parameter Kp, Ki dan Kd
5. Downloading program menuju Mikrokontroller
6. Dilakukan pengujian beban yang berbeda-beda

Pada pengujian sistem secara keseluruhan, langkah pertama yang harus dilakukan adalah merangkai peralatan dan rangkaian. Seperti pada Gambar 5.6



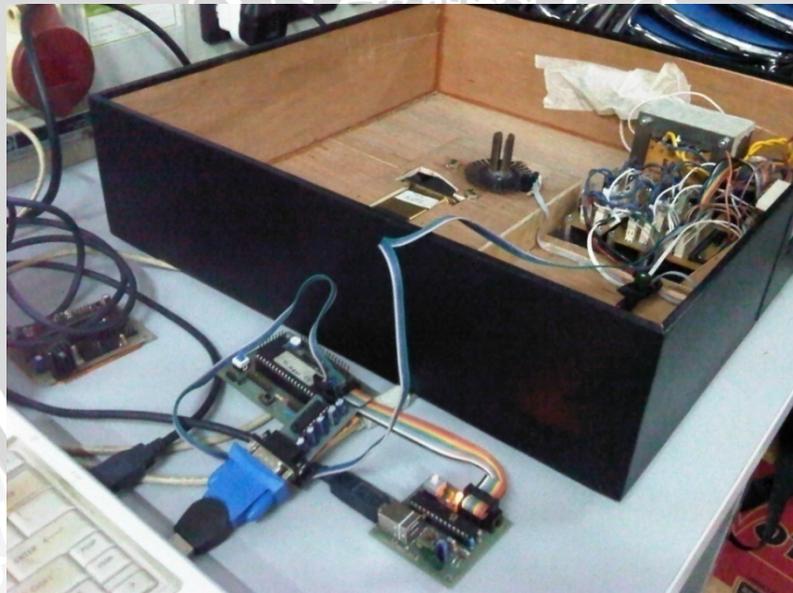
**Gambar 5.6** Blok Diagram Pengujian Sistem Secara Keseluruhan (perancangan)

Menghubungkan sensor pemutar gerabah yang akan memberikan instruksi sinyal pada pin Tx dan Rx yang terhubung oleh kabel serial RS-232. Kemudian tersambung juga oleh komputer melalui USB sehingga dapat diamati kecepatan berupa RPM setiap 1 detik yang dihasilkan. Seperti pada Gambar 5.7



**Gambar 5.7** Pengujian kecepatan menggunakan RS 232

Kemudian pada program CVAVR yang telah diprogram diberi masukan setpoint dengan nilai masing- masing 50 rpm untuk kecepatan lambat dan 100 rpm untuk kecepatan cepat. Pada alat ini menggunakan perangkat *writer* untuk memasukkan nilai setpoint dan nilai  $k_p$ ,  $k_i$ , dan  $k_d$ . Seperti pada Gambar 5.8



**Gambar 5.8** Penggunaan perangkat *writer*

Setelah itu akan diamati respon dari masing- masing setpoint ketika tanpa beban dan dengan beban 1 kg, 2 kg dan 3 kg, apakah sudah memenuhi *setpoint*.

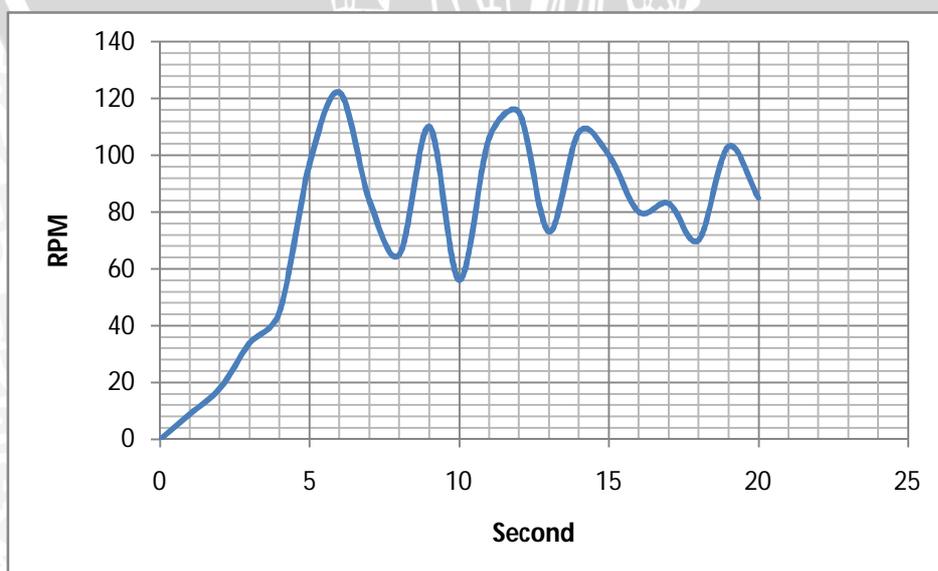
### 5.4.2.1 Pengujian kontroler PID

Tujuan dari pengujian ini adalah untuk mencari nilai parameter kontrol PID dengan menggunakan metode ke dua Osilasi Ziegler-Nichols dan untuk melihat apakah metode tersebut bisa diterapkan pada alat pemutar gerabah.

*Tahap-Tahap Pengujian PID:*

1. Mengatur nilai  $K_p$ ,  $K_i$  dan  $K_d$  mulai dari nol.
2. Menjalankan alat pemutar gerabah.
3. Mengamati tampilan grafik dari sensor *rotary encoder*.
4. Apabila grafik belum berbentuk osilasi kesinambungan, naikan nilai  $K_p$ , harga ditingkatkan dari nol ke suatu nilai kritis  $K_{cr}$ .
5. Apabila grafik sensor *rotary encoder* sudah membentuk osilasi kesinambungan, hitung nilai  $K_{cr}$  dan  $P_{cr}$ .
6. Hitung nilai  $T_i$  dan  $T_d$  dengan menggunakan nilai  $K_{cr}$  dan  $P_{cr}$ .
7. Hitung nilai  $K_i$  dan  $K_d$  dengan menggunakan nilai  $T_i$  dan  $T_d$ .
8. Dengan menggunakan nilai  $K_p$ ,  $K_i$  dan  $K_d$  yang telah didapat, amati kecepatan putaran alat pemutar gerabah.

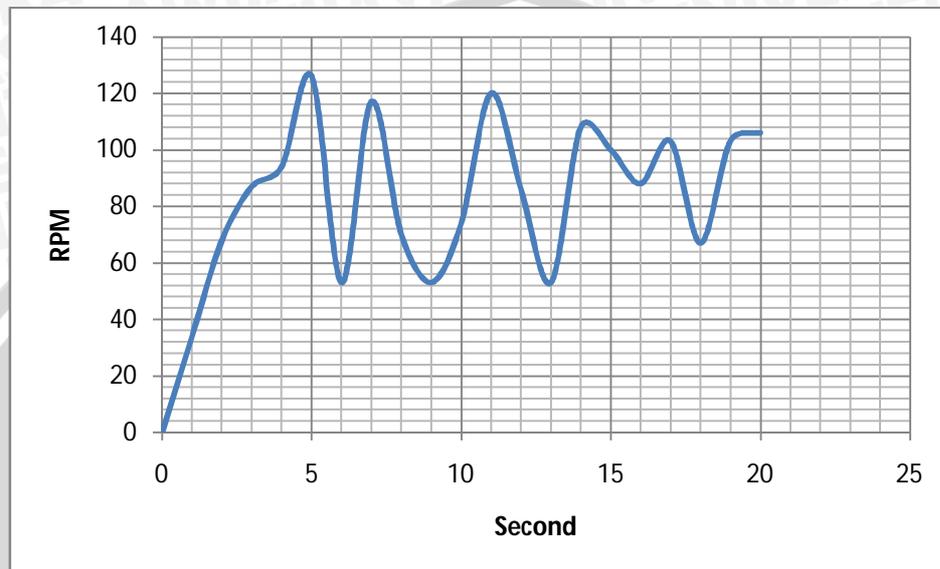
Pengujian terhadap kontrol kecepatan dengan metode osilasi Ziegler-Nichols dimulai dengan memberikan nilai 0 pada parameter  $T_i$  dan  $T_d$ . Sedangkan nilai  $K_p$  dinaikkan sedikit demi sedikit hingga didapatkan grafik yang berosilasi berkesinambungan. Hasil pengujian untuk respon kecepatan motor dengan menggunakan kontroler proporsional dengan nilai 1 ( $K_p=1$ ) dapat dilihat pada Gambar 5.9.



**Gambar 5.9** Grafik Respon kecepatan motor pemutar gerabah dengan  $K_p=1$

Terlihat di grafik bahwa respon kecepatan mengalami osilasi tetapi belum berkesinambungan sehingga masih perlu ditambahkan nilai  $K_p$ .

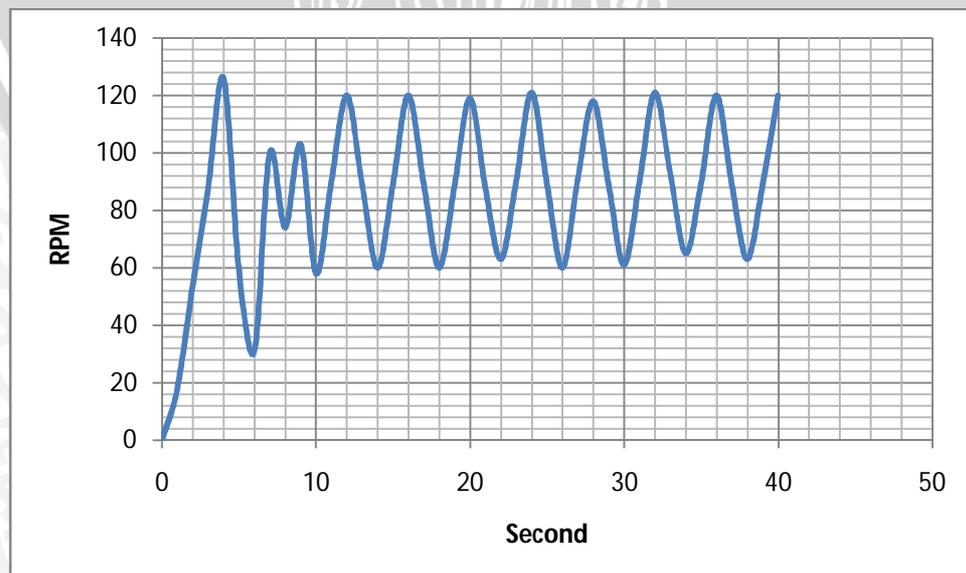
Hasil pengujian dengan menggunakan dengan nilai 2 ( $K_p=2$ ) dapat dilihat pada Gambar 5.10.



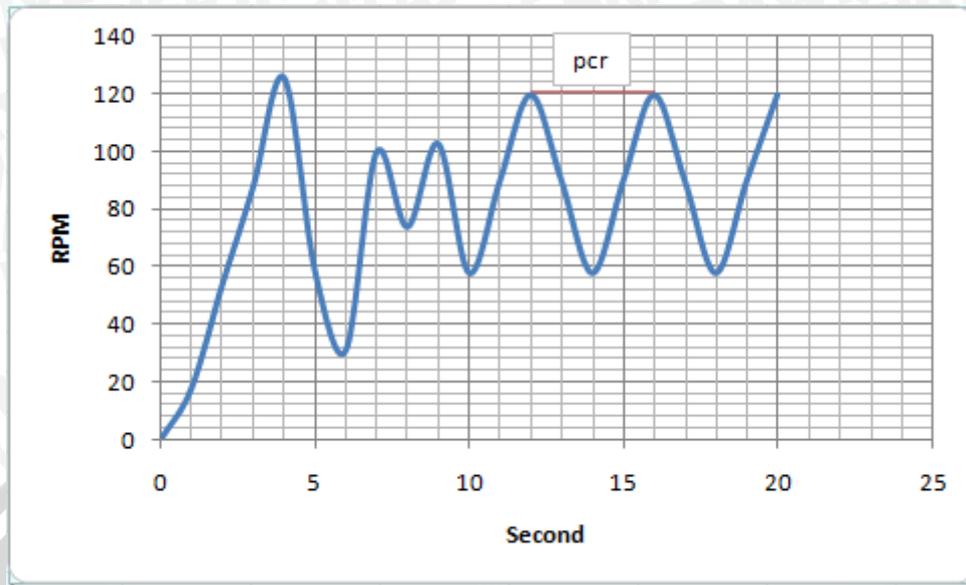
**Gambar 5.10** Grafik Respon kecepatan motor pemutar gerabah dengan  $K_p=2$

Pengujian dengan menggunakan nilai  $K_p=2$  terlihat bahwa respon sudah mulai mendekati kesinambungan dibandingkan  $k_p=1$  tetapi belum membentuk osilasi berkesinambungan.

Hasil pengujian untuk respon kecepatan motor dengan menggunakan kontroler proporsional dengan nilai 3 ( $K_p=3$ ) dapat dilihat pada Gambar 5.11.



**Gambar 5.11** Grafik Respon kecepatan motor pemutar gerabah saat terjadi osilasi kesinambungan dengan  $K_p= 3$



**Gambar 5.11** Grafik Respon kecepatan motor pemutar gerabah saat terjadi osilasi kesinambungan dengan  $K_p= 3$

Pada pengujian dengan menggunakan nilai  $K_p=3$  terlihat bahwa respon sudah mengalami osilasi kesinambungan pada detik ke 10. Terlihat bahwa pada saat kontroler proporsional bernilai 3 dapat membentuk osilasi berkesinambungan dibandingkan dengan  $k_p=2$  Sehingga dari grafik diatas dapat dihitung nilai  $K_{cr}$  dan  $P_{cr}$  yaitu,

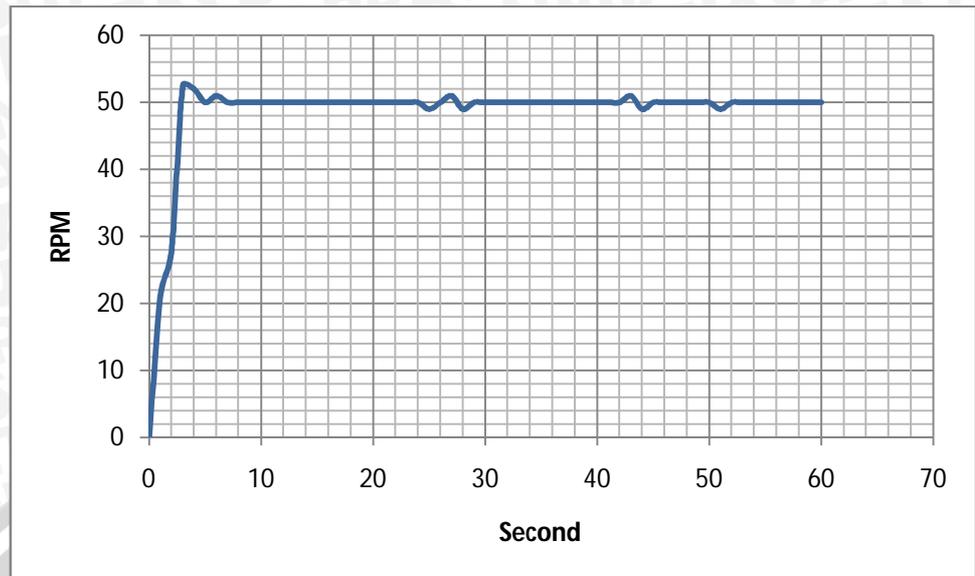
- $K_{cr} = 3$
- $P_{cr} = (16 - 12) \times \text{Time Sampling} = 4 \times 1 \text{ s} = 4 \text{ s}$
- $K_p = 0,6 \times K_{cr} = 0,6 \times 3 = 1,8$
- $T_i = 0,5 \times P_{cr} = 0,5 \times 4 \text{ s} = 2 \text{ s}$
- $T_d = 0,125 \times P_{cr} = 0,125 \times 4 \text{ s} = 0,5 \text{ s}$
- $K_i = \frac{K_p}{T_i} = \frac{1,8}{2} = 0,9$
- $K_d = K_p \times T_d = 1,8 \times 0,5 = 0,9$

Sehingga didapatkan nilai parameter  $K_p= 1,8$ ,  $K_i= 0,9$  dan  $K_d= 0,9$

### 5.4.3 Hasil Pengujian

#### 5.4.3.1 Set point 50 rpm

- Tanpa beban



**Gambar 5.12** Grafik Respon kecepatan motor pemutar gerabah pada set point 50 rpm tanpa beban

Dari gambar 5.12 didapatkan nilai *time settling* atau  $t_s$  dan *error steady state*, berikut akan dijelaskan pengertian *time settling* atau  $t_s$  dan *error steady state*, beserta perhitungannya.

a. *Time settling* atau  $t_s$  adalah waktu yang diperlukan kurva respon untuk mencapai dan menetap di daerah set point yang sudah ditentukan sebelumnya.

Pada kecepatan 50 rpm tanpa beban didapatkan  $T_s = 4s$

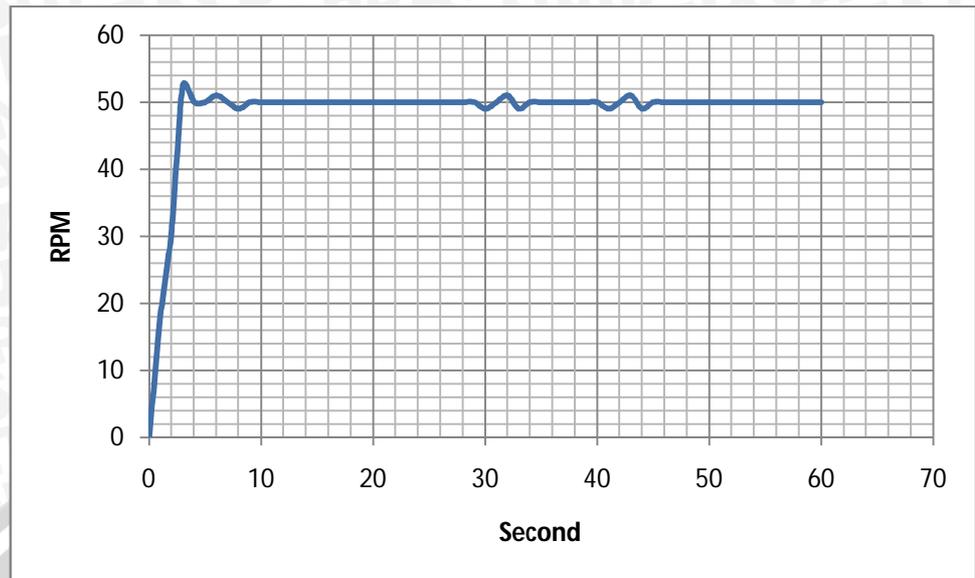
b. *Error steady state* adalah selisih antara nilai keluaran dengan nilai masukan pada saat kondisi *steady state*. Ess yang didapatkan dari pengujian dengan set point 50 rpm adalah:

$$ess = \frac{(51 - 50)}{50} \times 100\% = 2\%$$

c. *Maximum Overshoot* ( $M_p$ ) merupakan Nilai tertinggi dari grafik adalah 52 maka:

$$M_p = \frac{(52 - 50)}{50} \times 100\% = 4\%$$

- Beban 1 kg



**Gambar 5.13** Grafik Respon kecepatan motor pemutar gerabah pada set point 50 rpm  
1kg

Dari gambar 5.13 didapatkan nilai *time settling* atau  $t_s$  dan *error steady state*, berikut akan dijelaskan pengertian *time settling* atau  $t_s$  dan *error steady state*, beserta perhitungannya.

a. *Time settling* pada kecepatan 50 rpm dengan beban 1 kg didapatkan  $T_s = 4s$

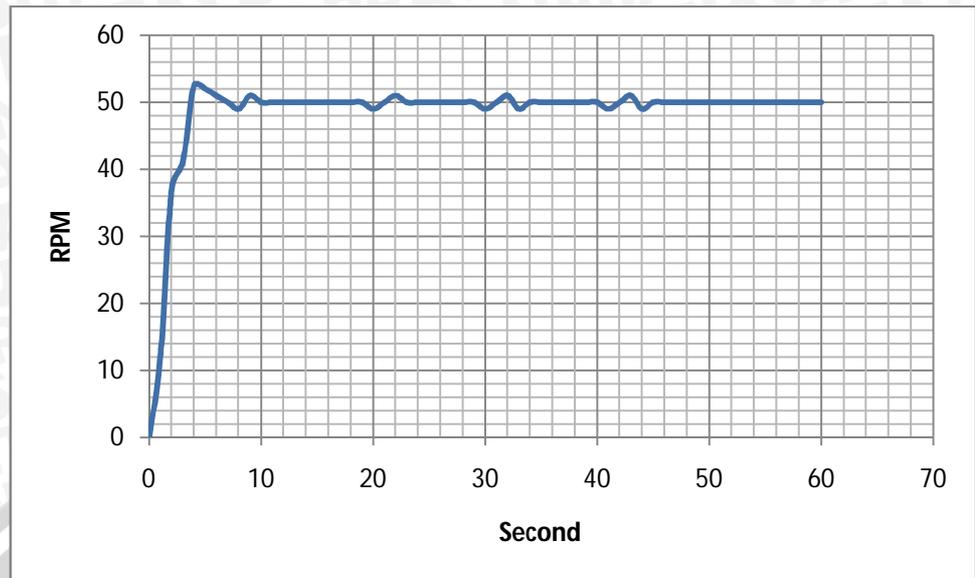
b. *Error steady state* yang didapatkan dari pengujian dengan set adalah:

$$ess = \frac{(51 - 50)}{50} \times 100\% = 2\%$$

c. *Maximum Overshoot* ( $M_p$ ) merupakan Nilai tertinggi dari grafik adalah 52, maka:

$$M_p = \frac{(52 - 50)}{50} \times 100\% = 4\%$$

- Beban 2 kg



**Gambar 5.14** Grafik Respon kecepatan motor pemutar gerabah pada set point 50 rpm 2kg

Dari gambar 5.14 didapatkan nilai *time settling* atau  $t_s$  dan *error steady state*, berikut akan dijelaskan pengertian *time settling* atau  $t_s$  dan *error steady state*, beserta perhitungannya.

a. *Time settling* pada kecepatan 50 rpm dengan beban 2 kg didapatkan  $T_s = 5s$

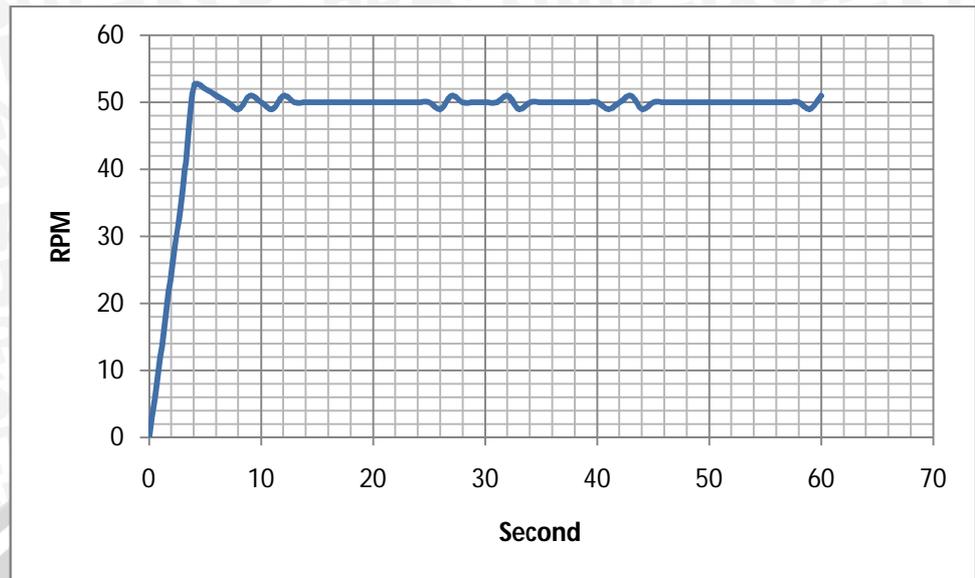
b. *Error steady state* yang didapatkan dari pengujian dengan set adalah:

$$ess = \frac{(51 - 50)}{50} \times 100\% = 2\%$$

c. *Maximum Overshoot* ( $M_p$ ) merupakan Nilai tertinggi dari grafik adalah 52, maka:

$$M_p = \frac{(52 - 50)}{50} \times 100\% = 4\%$$

- Beban 3 kg



**Gambar 5.15** Grafik Respon kecepatan motor pemutar gerabah pada set point 50 rpm 3kg

Dari gambar 5.15 didapatkan nilai *time settling* atau  $t_s$  dan *error steady state*, berikut akan dijelaskan pengertian *time settling* atau  $t_s$  dan *error steady state*, beserta perhitungannya.

a. *Time settling* pada kecepatan 50 rpm dengan beban 3 kg didapatkan  $T_s = 5s$

b. *Error steady state* yang didapatkan dari pengujian dengan set adalah:

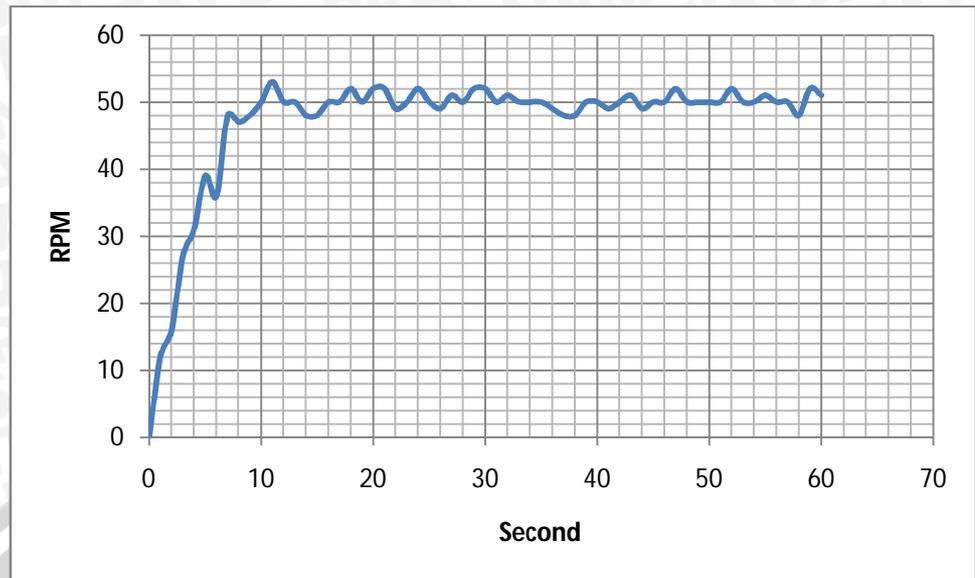
$$ess = \frac{(51 - 50)}{50} \times 100\% = 2\%$$

c. *Maximum Overshoot* ( $M_p$ ) merupakan Nilai tertinggi dari grafik adalah 52, maka:

$$M_p = \frac{(52 - 50)}{50} \times 100\% = 4\%$$

- Beban 4 kg





**Gambar 5.17** Grafik Respon kecepatan motor pemutar gerabah pada set point 50 rpm 5kg

Dari gambar 5.17 didapatkan nilai *time settling* atau  $t_s$  dan *error steady state*, berikut akan dijelaskan pengertian *time settling* atau  $t_s$  dan *error steady state*, beserta perhitungannya.

a. *Time settling* pada kecepatan 50 rpm dengan beban 5 kg didapatkan  $T_s = 11s$

b. *Error steady state* yang didapatkan dari pengujian dengan set adalah:

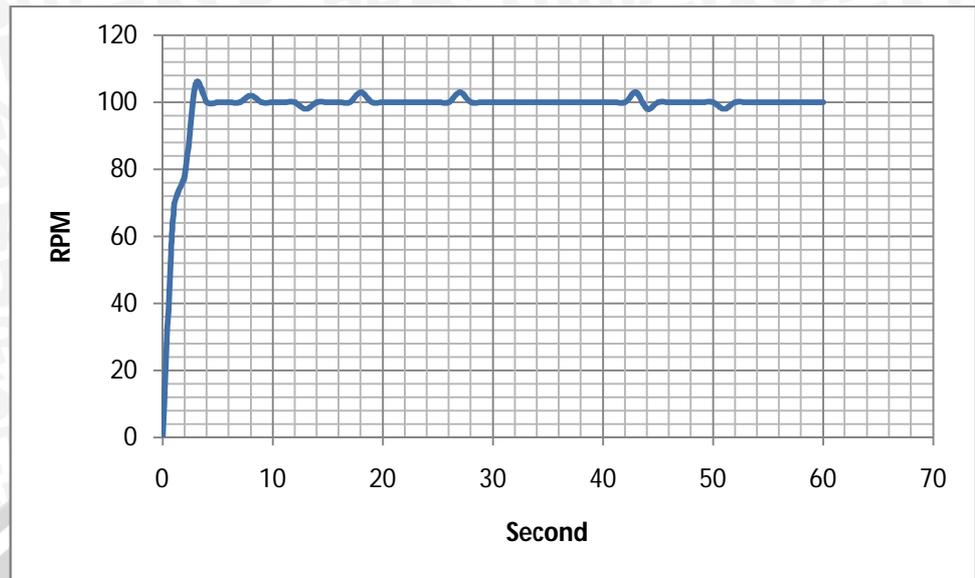
$$ess = \frac{(52 - 50)}{50} \times 100\% = 4\%$$

c. *Maximum Overshoot* ( $M_p$ ) merupakan Nilai tertinggi dari grafik adalah 53, maka:

$$M_p = \frac{(53 - 50)}{50} \times 100\% = 6\%$$

#### 5.4.3.2 Set point 100 rpm

- Tanpa beban



**Gambar 5.18** Grafik Respon kecepatan motor pemutar gerabah pada set point 100 rpm tanpa beban

Dari gambar 5.18 didapatkan nilai *time settling* atau  $t_s$  dan *error steady state*, berikut akan dijelaskan pengertian *time settling* atau  $t_s$  dan *error steady state*, beserta perhitungannya.

a. *Time settling* pada kecepatan 100 rpm tanpa beban didapatkan  $T_s = 4s$

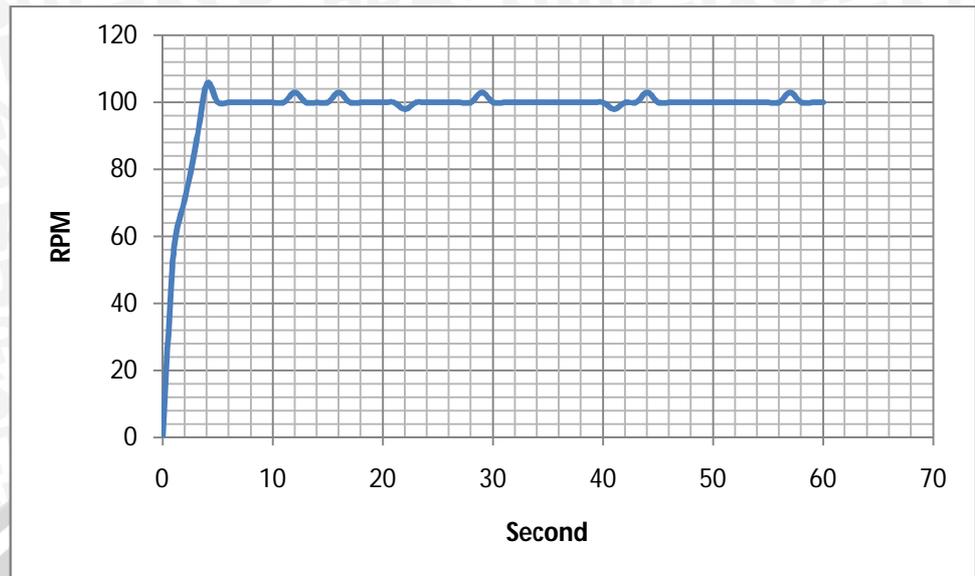
b. *Error steady state* yang didapatkan dari pengujian dengan set adalah:

$$ess = \frac{(103 - 100)}{100} \times 100\% = 3\%$$

c. *Maximum Overshoot* ( $M_p$ ) merupakan Nilai tertinggi dari grafik adalah 105, maka:

$$M_p = \frac{(105 - 100)}{100} \times 100\% = 5\%$$

- Beban 1 kg



**Gambar 5.19** Grafik Respon kecepatan motor pemutar gerabah pada set point 100 rpm 1kg

Dari gambar 5.19 didapatkan nilai *time settling* atau  $t_s$  dan *error steady state*, berikut akan dijelaskan pengertian *time settling* atau  $t_s$  dan *error steady state*, beserta perhitungannya.

a. *Time settling* pada kecepatan 100 rpm dengan beban 1 kg didapatkan  $T_s = 4s$

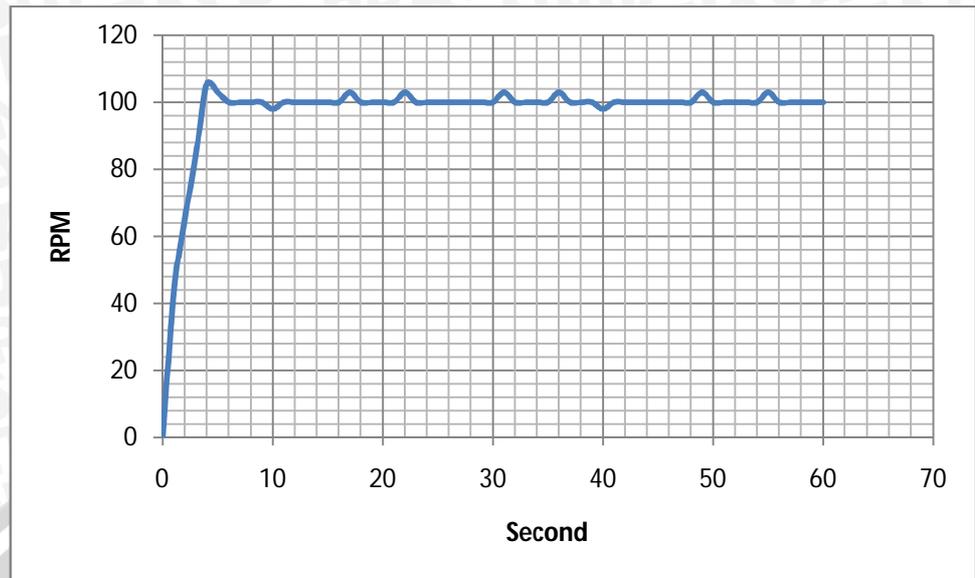
b. *Error steady state* yang didapatkan dari pengujian dengan set adalah:

$$ess = \frac{(103 - 100)}{100} \times 100\% = 3\%$$

c. *Maximum Overshoot* ( $M_p$ ) merupakan Nilai tertinggi dari grafik adalah 105, maka:

$$M_p = \frac{(105 - 100)}{100} \times 100\% = 5\%$$

- Beban 2 kg



**Gambar 5.20** Grafik Respon kecepatan motor pemutar gerabah pada set point 100 rpm 2kg

Dari gambar 5.20 didapatkan nilai *time settling* atau  $t_s$  dan *error steady state*, berikut akan dijelaskan pengertian *time settling* atau  $t_s$  dan *error steady state*, beserta perhitungannya.

a. *Time settling* pada kecepatan 100 rpm dengan beban 2 kg didapatkan  $T_s = 5s$

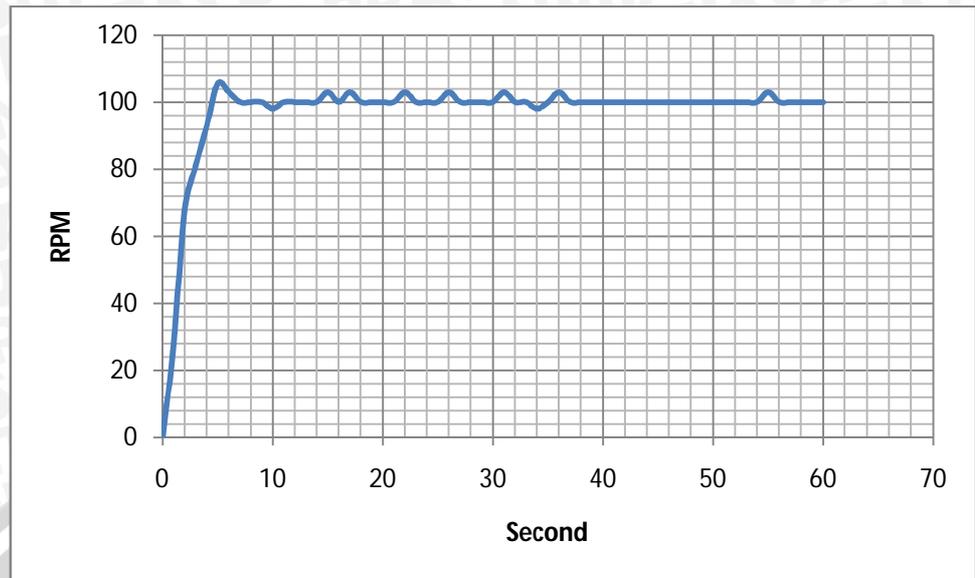
b. *Error steady state* yang didapatkan dari pengujian dengan set adalah:

$$ess = \frac{(103 - 100)}{100} \times 100\% = 3\%$$

c. *Maximum Overshoot* ( $M_p$ ) merupakan Nilai tertinggi dari grafik adalah 105, maka:

$$M_p = \frac{(105 - 100)}{100} \times 100\% = 5\%$$

- Beban 3 kg



**Gambar 5.21** Grafik Respon kecepatan motor pemutar gerabah pada set point 100 rpm 3kg

Dari gambar 5.21 didapatkan nilai *time settling* atau  $t_s$  dan *error steady state*, berikut akan dijelaskan pengertian *time settling* atau  $t_s$  dan *error steady state*, beserta perhitungannya.

a. *Time settling* pada kecepatan 100 rpm dengan beban 3 kg didapatkan  $T_s = 5s$

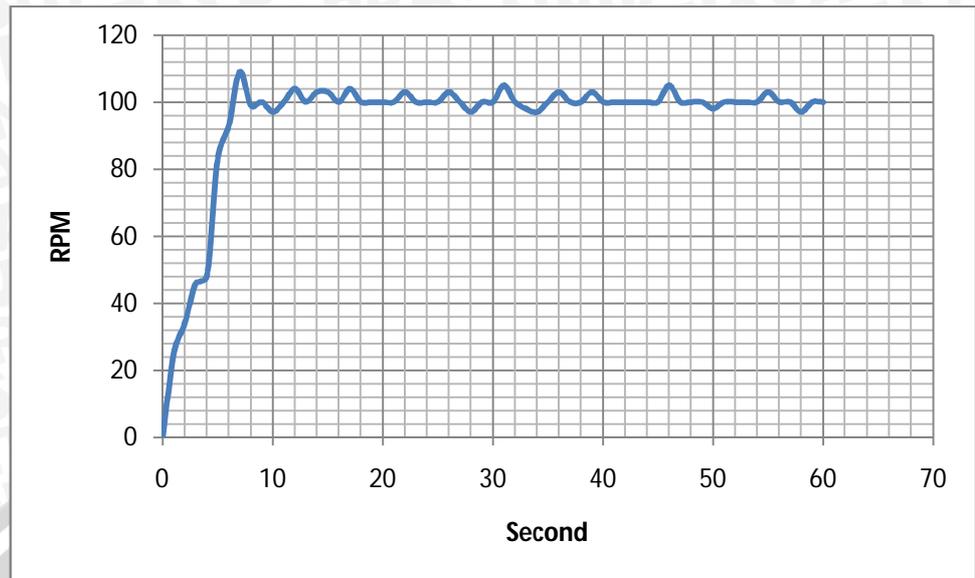
b. *Error steady state* yang didapatkan dari pengujian dengan set adalah:

$$ess = \frac{(103 - 100)}{100} \times 100\% = 3\%$$

c. *Maximum Overshoot* ( $M_p$ ) merupakan Nilai tertinggi dari grafik adalah 105, maka:

$$M_p = \frac{(105 - 100)}{100} \times 100\% = 5\%$$

- Beban 4 kg



**Gambar 5.22** Grafik Respon kecepatan motor pemutar gerabah pada set point 100 rpm 4kg

Dari gambar 5.22 didapatkan nilai *time settling* atau  $t_s$  dan *error steady state*, berikut akan dijelaskan pengertian *time settling* atau  $t_s$  dan *error steady state*, beserta perhitungannya.

a. *Time settling* pada kecepatan 100 rpm dengan beban 4 kg didapatkan  $T_s = 7s$

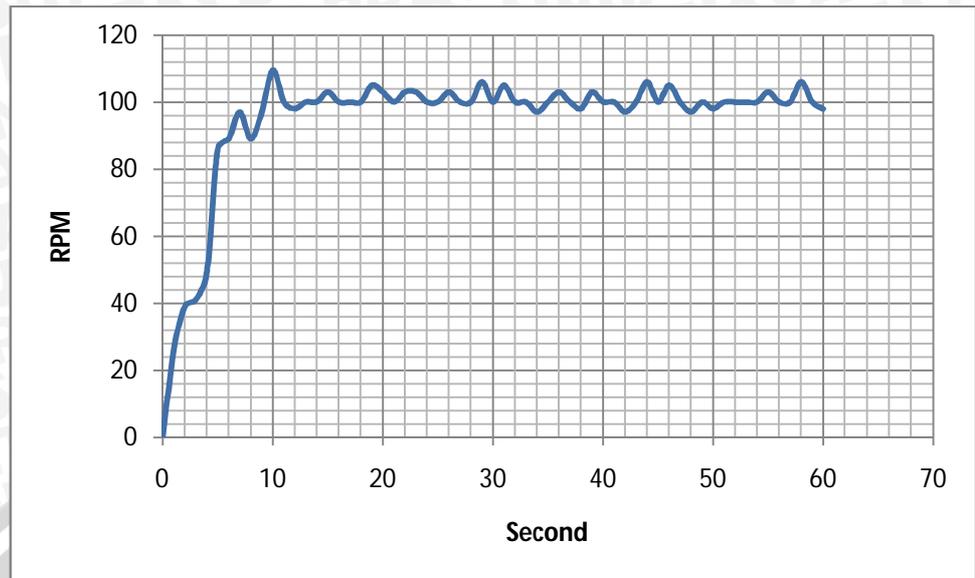
b. *Error steady state* yang didapatkan dari pengujian dengan set adalah:

$$ess = \frac{(104 - 100)}{100} \times 100\% = 4\%$$

c. *Maximum Overshoot* ( $M_p$ ) merupakan Nilai tertinggi dari grafik adalah 106, maka:

$$M_p = \frac{(106 - 100)}{100} \times 100\% = 6\%$$

- Beban 5 kg



**Gambar 5.23** Grafik Respon kecepatan motor pemutar gerabah pada set point 100 rpm 5kg

Dari gambar 5.23 didapatkan nilai *time settling* atau  $t_s$  dan *error steady state*, berikut akan dijelaskan pengertian *time settling* atau  $t_s$  dan *error steady state*, beserta perhitungannya.

a. *Time settling* pada kecepatan 100 rpm dengan beban 5 kg didapatkan  $T_s = 10s$

b. *Error steady state* yang didapatkan dari pengujian dengan set adalah:

$$ess = \frac{(105 - 100)}{100} \times 100\% = 5\%$$

c. *Maximum Overshoot* ( $M_p$ ) merupakan Nilai tertinggi dari grafik adalah 107, maka:

$$M_p = \frac{(107 - 100)}{100} \times 100\% = 7\%$$

Pengujian ini ditentukan memiliki prosentase kesalahan Ess yang diperbolehkan adalah 5% berdasarkan sistem yang dinyatakan. Dengan demikian nilai toleransi yang diperbolehkan pada kecepatan 50 rpm adalah dalam kisaran 2 sampai 3 rpm dan pada 100 rpm adalah 5 rpm. Berikut merupakan data hasil pengujian performansi sistem pada Tabel 5.3 dan 5.4.

**Tabel 5.3** Tabel Data Performansi Sistem set point 50 rpm

Beban (kg)	Motor (rpm)	Settling time (ts)	Overshoot (%)	Ess (%)
0	50	4 second	4	2
1	50	4 second	4	2
2	50	5 second	4	2
3	50	5 second	4	2
4	50	8 second	6	4
5	50	11 second	6	4

**Tabel 5.4** Tabel Data Performansi Sistem set point 100 rpm

Beban (kg)	Motor (rpm)	Settling time (ts)	Overshoot (%)	Ess (%)
0	100	4 second	5	3
1	100	4 second	5	3
2	100	5 second	5	3
3	100	5 second	5	3
4	100	7 second	6	4
5	100	10 second	7	5

Berikut merupakan hasil dari pengujian yang telah dilakukan pada skripsi sebelumnya (Rahma (2012)) yang menggunakan kontrol logika fuzzy pada tabel 5.5

**Tabel 5.5** Tabel Data Performansi Sistem Menggunakan Kontrol Logika Fuzzy

NO.	SETPOINT (rpm)	BEBAN							
		TANPA BEBAN		1 KG		1,5 KG		2 KG	
		$t_s$ (second)	ess (%)	$t_s$ (second)	ess (%)	$t_s$ (second)	ess (%)	$t_s$ (second)	ess (%)
1.	33	78	3,39	85	2,27	95	2,27	68	2,27

2.	52	68	0,96	34	4,55	53	4,55	36	0,96
3.	65	67	0,96	55	0,96	34	0,96	61	3,84
4.	82	47	1,67	40	1,67	50	1,67	47	0,6
5.	95	43	1,31	49	1,31	49	1,31	61	1,31

Jadi hasil pengujian untuk alat pemutar gerabah dapat dibandingkan dari tabel 5.4 dan tabel 5.5 bahwa dengan menggunakan metode Osilasi Ziegler Nichols didapatkan  $t_s$  yang lebih cepat dibandingkan dengan menggunakan kontrol logika fuzzy.



## 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:

Hasil pengujian kontrol PID menunjukkan bahwa PID Ziegler Nichols metode osilasi dapat digunakan sebagai kontrol kecepatan motor DC pada alat pemutar gerabah serta memberikan nilai parameter yang sesuai dengan sistem dan menghasilkan respon yang lebih cepat dibandingkan dengan kontrol logika fuzzy untuk mencapai keadaan steady state. Parameter PID yang didapatkan adalah ( $K_p=1,8$ ,  $K_i=0,9$ ,  $K_d=0,9$ ). Untuk kecepatan 50 rpm dan 100 rpm,  $t_s$  yang didapat sekitar 4 second sampai 11 second, *Maximum Overshoot* sekitar 4% sampai 7%, dan *Error steady state* sekitar 2% sampai 5 %.

#### 6.2 Saran

Beberapa hal yang direkomendasikan untuk pengembangan lebih lanjut adalah:

1. Penggunaan sensor *rotary encoder* dengan jumlah lubang yang lebih banyak agar kesalahan pengukuran akan semakin kecil dan pengukuran kecepatan lebih tepat lagi.
2. Pembuatan mekanik yang baik dan lebih presisi akan membuat alat pemutar gerabah semakin stabil dalam pergerakan dan kecepatannya.
3. Penambahan sensor berat pada alat pemutar gerabah akan memudahkan dalam pengambilan data untuk pengembangan penelitian selanjutnya.

## DAFTAR PUSTAKA

Amalia Nur, Rahma. 2012. *Pemutar Gerabah Dengan Kontrol Logika Fuzzy Takagi-Sugeno Kang Berbasis Mikrokontroler ATMEGA 8535* Malang: Universitas Brawijaya.

Atmel. 2008. *ATMEGA32/ATMEGA32L, 8-bit AVR with 32 Kbytes in System ProgramableFlash*.<http://www.atmel.com/products/microcontrollers/avr/default.aspx?tab=documents>

Elmiawan Akbar, Arnas. 2013. *Implementasi Sistem Navigasi Wall Following Menggunakan Kontroler PID dengan Metode Tuning pada Robot Kontes Robot Cerdas Indonesia (KRCI)*. Malang: Universitas Brawijaya.

Ogata, Katsuhiko. 1997. *Teknik Kontrol Automatik Jilid I*. Jakarta. Penerbit Erlangga

Ogata, Katsuhiko. 1997. *Teknik Kontrol Automatik Jilid II*. Jakarta. Penerbit Erlangga

<http://id.wikipedia.org/wiki/Tembikar/Gerabah>. diakses 23 Agustus 2012

