

**SISTEM PENGENDALIAN KECEPATAN MOTOR PEMOTONG
KAYU MENGGUNAKAN METODE PID BERBASIS ARDUINO UNO**

SKRIPSI

TEKNIK ELEKTRO KONSENTRASI TEKNIK KONTROL

Ditujukan untuk memenuhi persyaratan
Memperoleh gelar Sarjana Teknik



RIDHO ROBBY ISRONI
NIM. 0910633069 - 63

KEMENTERIAN RISET TEKNOLOGI DAN PENDIDIKAN TINGGI
UNIVERSITAS BRAWIJAYA
FAKULTAS TEKNIK
JURUSAN TEKNIK ELEKTRO
MALANG
2016

UNIVERSITAS BRAWIJAYA



LEMBAR PENGESAHAN

SISTEM PENGENDALIAN KECEPATAN MOTOR PEMOTONG KAYU MENGGUNAKAN METODE PID BERBASIS ARDUINO UNO

SKRIPSI

TEKNIK ELEKTRO KONSENTRASI TEKNIK KONTROL

Ditujukan untuk memenuhi persyaratan

Memperoleh gelar Sarjana Teknik



RIDHO ROBBY ISRONI
NIM. 0910633069 – 63

Skripsi ini telah direvisi dan disetujui oleh dosen pembimbing
Pada tanggal 20 Juni 2016

Dosen Pembimbing I

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

Dosen Pembimbing II

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

Mengetahui
Ketua Jurusan/Ketua Program Studi

M. Aziz Muslim, S.T., M.T., Ph.D.
NIP. 19741203 200012 1 001

UNIVERSITAS BRAWIJAYA



JUDUL SKRIPSI :

**SISTEM PENGENDALIAN KECEPATAN MOTOR PEMOTONG KAYU
MENGUNAKAN METODE PID BERBASIS ARDUINO UNO**

Nama Mahasiswa : Ridho Robby Isoni
 NIM : 0910633069
 Program Studi : Teknik Elektro
 Minat : Teknik Kontrol

KOMISI PEMBIMBING :

Ketua : Ir. Purwanto, M.T.

Anggota : Ir. Retnowati, M.T.

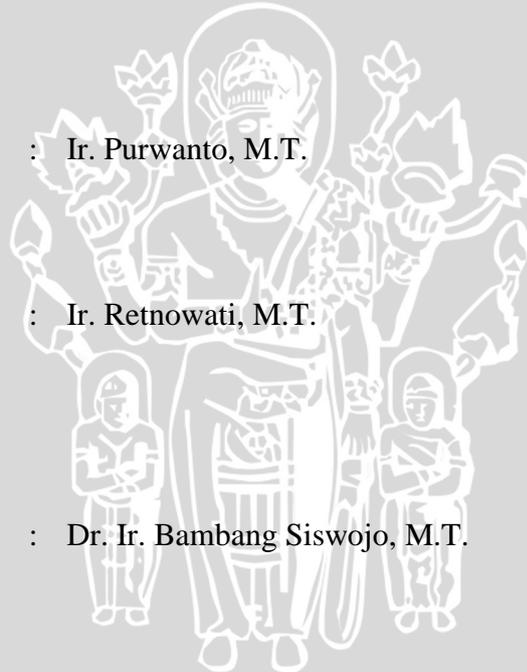
TIM DOSEN PENGUJI :

Dosen Penguji 1 : Dr. Ir. Bambang Siswojo, M.T.

Dosen Penguji 2 : Goegoes Dwi N., S.T., M.T.

Dosen Penguji 3 : Rahmadwati, S.T., M.T., Ph.D.

SK Penguji : 632/ UN10.6/ SK/ 2016



PERNYATAAN ORISINALITAS SKRIPSI

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya dan berdasarkan hasil penelusuran berbagai karya ilmiah, gagasan dan masalah ilmiah yang diteliti dan diulas dalam naskah Skripsi ini adalah asli dari pemikiran saya. Tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata di dalam naskah Skripsi ini dapat dibuktikan terdapat unsur-unsur jiplakan, saya bersedia Skripsi ini dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, pasal 25 ayat 2, dan pasal 70)

Malang, 26 Januari 2016

Mahasiswa,

Materai Rp. 6000,-

Ridho Robby Isoni

0910633069



UNIVERSITAS BRAWIJAYA



*Teriring Ucapan Terima Kasih kepada:
Ayahanda dan Ibunda tercinta*

UNIVERSITAS BRAWIJAYA



PENGANTAR

Alhamdulillah, puji syukur kehadiran Allah Subhanahu Wa Ta'ala yang telah memberikan Rahmat dan Karunia-Nya kepada saya selaku penulis skripsi dengan judul: Sistem pengendalian Kecepatan Motor Pemotong Kayu Menggunakan Metode PID Berbasis Arduino Uno. Karena hanya Dialah yang dapat memberikan kekuatan dan kesempatan sehingga skripsi ini dapat terselesaikan. Tak lupa Shalawat serta salam selalu turunkan kepada suri tauladan kita Nabi Muhammad Shalallahu Alaihi Wa Salam.

Terselesainya skripsi ini tak lepas dari campur tangan dan bantuan dari banyak pihak, sehingga dalam kesempatan yang baik ini saya ingin mengemukakan rasa terima kasih dan penghargaan yang sebesar besarnya kepada:

1. Kedua orang tua Herry Santoso (Alm) dan Nurul Musyafa'ah yang tiada hentinya memberikan inspirasi dan semangat dalam penulisan penelitian ini. Serta kepada kakak Femmy Dianita Oktora dan Adik M. Afrizal Setiawan yang memberikan banyak bantuan.
2. Bapak M. Aziz Muslim, S.T., M.T., Ph.D selaku Ketua Jurusan Teknik Elektro Universitas Brawijaya.
3. Bapak Hadi Suyono, S.T., M.T., Ph.D selaku Sekretaris Jurusan Teknik Elektro Universitas Brawijaya.
4. Bapak Ir. Purwanto, M.T selaku KKDK Teknik Kontrol dosen pembimbing skripsi yang telah banyak memberikan pengarahan, bimbingan, nasehat, saran dan motivasinya.
5. Ibu Ir. Retnowati, M.T selaku dosen pembimbing skripsi juga yang telah memberikan banyak pengarahan, bimbingan, nasehat, saran dan motivasinya.
6. Ibu Ir. Nurussa'adah, M.T selaku dosen pembimbing akademik yang selalu memberikan nasehat-nasehat terbaiknya, bimbingan, saran dan motivasinya.
7. Teman-teman dekat dan orang-orang yang terlibat secara tidak langsung dalam proses penyusunan, yang tidak dapat disebutkan satu persatu.

Semoga Allah SWT membalas amalan ikhlas kami dan semua pihak yang turut membantu sehingga skripsi ini terselesaikan. Akhirnya, kami menyadari bahwa skripsi ini masih jauh dari sempurna namun semoga skripsi ini dapat memberikan manfaat bagi kita semua kedepannya. Aamiin Allahumma Aamiin.

Malang, 29 Agustus 2015

Penulis



DAFTAR ISI

PENGANTAR	i
DAFTAR ISI	iii
DAFTAR GAMBAR.....	vi
DAFTAR TABEL	viii
RINGKASAN.....	ix
<i>SUMMARY</i>	ix
BAB I.....	1
PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Batasan Masalah.....	2
1.4. Tujuan	2
1.5. Sistematika Pembahasan.....	3
BAB II	4
TINJAUAN PUSTAKA	4
2.1. Motor.....	4
2.2. Sensor Optocoupler	5
2.3. Kontroler	6
2.3.1. Kontroler Proporsional	7
2.3.2. Kontroler Integral.....	8
2.3.3. Kontroler Diferensial.....	8
2.3.4. Kontroler Proporsional Integral Diferensial (PID).....	9
2.3.5. Metode Kontroler Proporsional Integral Diferensial (PID).....	10
2.4. Komunikasi Serial RS232	14
2.5. Arduino Uno.....	15
2.5.1. Daya.....	16

2.5.2.	Memori.....	17
2.5.3.	Input dan Output.....	17
2.5.4.	Komunikasi	18
BAB III.....		19
METODOLOGI PENELITIAN		19
3.1.	Studi Literatur.....	19
3.2.	Perancangan dan Pembuatan Alat.....	19
3.2.1.	Perancangan dan Pembuatan Perangkat Keras (<i>Hardware</i>).....	19
3.2.2.	Perancangan dan Pembuatan Perangkat Lunak (<i>Software</i>)	19
3.3.	Pengujian Alat	20
3.3.1.	Pengujian Tiap Blok.....	20
3.3.2.	Pengujian Keseluruhan Sistem	20
3.4.	Pengambilan Kesimpulan	20
BAB IV		21
PERANCANGAN & PEMBUATAN ALAT		21
4.1.	Diagram Blok Sistem	21
4.2.	Prinsip Kerja Alat	22
4.3.	Perancangan Perangkat Keras.....	22
4.3.1.	Perancangan Mekanik	22
4.3.2.	Perancangan Sensor Optocoupler	23
4.3.3.	Perancangan Rangkaian Pengatur Kecepatan Motor Gergaji.....	24
4.3.4.	Pembuatan Rangkaian Pengatur Arah Putaran Motor DC	25
4.3.5.	Konfigurasi Arduino Uno	26
4.4.	Perancangan Perangkat Lunak.....	27
4.4.1.	Perancangan Kontroler PID	27
4.4.2.	Flowchart.....	36
BAB V.....		37
PENGUJIAN & ANALISIS SISTEM		37
5.1.	Pengujian Sensor Kecepatan.....	37

5.1.1.	Peralatan yang digunakan.....	37
5.1.2.	Prosedur Pengujian.....	37
5.1.3.	Hasil Pengujian	38
5.2.	Pengujian Motor Gergaji	39
5.2.1.	Peralatan yang digunakan.....	39
5.2.2.	Prosedur Pengujian.....	40
5.2.3.	Hasil Pengujian	40
5.3.	Pengujian Blok Kontroler	41
5.3.1.	Pengujian tanpa Kontroler.....	41
5.3.2.	Pengujian dengan Kontroler	47
5.4.	Pengujian Keseluruhan Sistem	54
BAB VI	67
PENUTUP	67
6.1.	Kesimpulan	67
DAFTAR PUSTAKA	68
LAMPIRAN I	69
LAMPIRAN II	71
LAMPIRAN III	86



DAFTAR GAMBAR

Gambar 2.1 Hitachi Koki C7SS	4
Gambar 2.2 Sensor Optocoupler	5
Gambar 2.3 Diagram Blok Kontroler Proporsional	8
Gambar 2.4 Diagram Blok Kontroler Integral	8
Gambar 2.5 Diagram Blok Kontroler Diferensial	9
Gambar 2.6 Diagram Blok Kontroler PID	10
Gambar 2.7 Kurva Respon Unit Step yang Menunjukkan 25% Maximum Overshoot	10
Gambar 2.8 Respon Plant Terhadap Masukan Berupa Unit Step	11
Gambar 2.9 Kurva Respon yang Berbentuk S	11
Gambar 2.10 Sistem Loop Tertutup dengan Kontroler Proporsional	13
Gambar 2.11 Osilasi Berkesinambungan dengan Periode Pcr	13
Gambar 2.12. Tampak depan Arduino Uno	16
Gambar 4.1 Diagram Blok Sistem Pengaturan	21
Gambar 4.2 Rangkaian Sensor Optocoupler	23
Gambar 4.3 Desain Umum Optocoupler	24
Gambar 4.4 Rangkaian Pengaturan Kecepatan Motor Gergaji	24
Gambar 4.5 Schematic Pengaturan Arah Putaran Motor Penggerak Potensio	25
Gambar 4.6 Konfigurasi microcontroller Arduino Uno	26
Gambar 4.7 Respon PID Zyglar Nichols	27
Gambar 4.8 Respon Sistem dengan Kp, Ki, dan Kd Berdasarkan Ziegler-Nichols	29
Gambar 4.9 Grafik Respon untuk Kp = 0.1	30
Gambar 4.10 Grafik Respon untuk Kp = 0.3	31
Gambar 4.11 Grafik Respon untuk Kp = 0.4	31
Gambar 4.12 Grafik Respon untuk Kp = 0.4 dan Ki= 0.1	32
Gambar 4.13 Grafik Respon untuk Kp = 0.4 dan Ki = 0.3	33
Gambar 4.14 Grafik Respon untuk Kp = 0.4 dan Ki = 0.5	33
Gambar 4.15 Grafik Respon untuk Kp = 0.95, Ki = 0.1, dan Kd = 0.1	34
Gambar 4.16 Grafik Respon untuk Kp = 0.4, Ki = 0.1, dan Kd = 0.5	35
Gambar 4.17 Flowchart Program	36
Gambar 5.1 Skema pengujian sensor Optocoupler	38



Gambar 5.2 Grafik Perbandingan Nilai Pembacaan Sensor Kecepatan dengan Data Praktek39

Gambar 5.3 Rangkaian Pengujian Kecepatan Motor.....40

Gambar 5.4 Grafik Hubungan Tengan dengan Kecepatan Motor.....41

Gambar 5.5 Grafik Pengujian tanpa Kontroler47

Gambar 5.6 Grafik Pengujian dengan Kontroler54

Gambar 5.7 Data Grafik Pengujian Berbeban64



DAFTAR TABEL

Tabel 2.1 Aturan Penalaan Ziegler-Nichols Berdasarkan Respon Unit Step Dari Plan8.....	12
Tabel 2.2 Aturan Dasar Ziegler-Nichols Berdasarkan Critical Gain Kcr dan Critical Period Pcr ...	13
Tabel 4.1 Deskripsi konfigurasi Arduino Uno dengan sensordan motor DC	26
Tabel 4.2 Dasar Ziegler-Nichols Berdasarkan <i>Critical Gain</i> Kcr dan <i>Critical Period</i> Pcr	28
Tabel 5.1 Data Hasil Pengujian Sensor Optocoupler	38
Tabel 5.2 Data Pengujian tanpa Kontroler.....	41
Tabel 5.3 Data Pengujian Menggunakan Kontroler.....	47
Tabel 5.4 Data Pengujian Keseluruhan dengan Beban 3 Jenis Kayu	55



RINGKASAN

Ridho Robby Isoni, Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya, Desember 2015, *Sistem Pengendalian Kecepatan Motor Pemotong Kayu Menggunakan Metode PID Berbasis Arduino Uno*, Dosen Pembimbing : Purwanto dan. Retnowati.

Di Indonesia terdapat sekitar 4000 jenis pohon yang berpotensi dapat dimanfaatkan kayunya oleh manusia. Sekian banyak jenis pohon pastinya memiliki karakteristik kayu yang berbeda beda, karakteristik yang dimaksud disini adalah tingkat kekerasan kayunya. Tingkat kekerasan kayu sangat berpengaruh pada lamanya proses pemotongan.

Pada penelitian ini dilakukan pemotongan 3 jenis kayu dipilih dari sekian banyak jenis kayu karena sudah mewakili 3 tingkat kekerasan kayu dengan ketebalan maksimal 10 cm. Kayu- kayu tersebut yaitu kayu Jati mewakili jenis kayu yang cenderung keras, kayu Mahoni yang mewakili jenis kayu dengan tingkat kekerasan sedang, dan kayu Albasia mewakili jenis kayu yang cenderung tidak keras. Penggunaan metode PID (*Proporsional Integral Derivative*) bertujuan untuk mempertahankan kecepatan motor pemotong kayu agar apabila diberi beban kayu dengan tingkat kekerasan berbeda motor dapat memotong dengan kecepatan yang relatif sama sehingga lamanya waktu pemotongan tiap jenis kayu relatif sama pula, serta nantinya diharapkan dapat mempercepat proses pemotongan. Dengan kontroler PID diharap mendapatkan respon kecepatan terbaik.

Dari hasil penelitian respon sistem untuk mencapai *steady* pada 3.36 detik. nilai parameter $K_p=0.4$, $K_i=0.1$, $K_d=0.5$, dan persen *Ess* (*error steady state*) untuk beban kayu Albasia sebesar 0.930222%, untuk beban kayu Mahoni sebesar 1.056311% untuk beban kayu Jati sebesar 0.685%. Selain itu juga didapat persen M_p (*maximum overshoot*) untuk beban kayu Albasia sebesar 3.222% untuk beban kayu Mahoni sebesar 2.222% untuk beban kayu Jati sebesar 2.444%.

Kata Kunci : kayu, pemotong kayu, kontroler PID, kecepatan motor

SUMMARY

Ridho Robby Isoni, Departement of Electrical Engineering, Faculty of Engineering, Universitas Brawijaya, Desember 2015, *Motor Speed Control System for Wood - Cutter Based PID Method Using Arduino Uno, Supervisor* : Purwanto and Retnowati.

In Indonesia there are about 4000 species of tress that could potentially be utilized wood by humans. Certainly many types of tress have different characteristics of different wood, the characteristics referred to here is the level of hardness timber. The level of hardness is very influential in this old wood cutting process. In this research, cutting 3 types of wood selected from many types of wood because it's represents three levels of hardness's wood with a maximum thickness of 10 cm. The Timbers are rosewood represents the type of wood that tend to be hard, Mahogany wood which represent the type of wood with moderate hardness level, and Albasia wood represent the type of wood that tend to be hard. The used of method PID (Proportional Integral Derivative) aims to maintain the motor speed cutters that speed is relatively same, so the length of time cutting each type of wood is relatively same as well, and is expected to get the best speed of respond. The result of research shows that to achieve steady syste respone at 3 : 36 seconds. Parameter value of $K_p = 0,4$; $K_i = 0,1$; $K_d = 0,5$, and percentage ESS (steady state error) for a Albasia wood is 0,930222%, Mahogany is 1,056311% , and for Teakwood is 0,685%. It's also obtained percentage M_p (maximum overshoot) to Albasia timber load is 3,222%, Mahogany 2,222%, and Teak wood 2,444%.

Keywords : Wood, Wood cutters, PID controlers, motor speed

BAB I

PENDAHULUAN

1.1.Latar Belakang

Meskipun Kepulauan Indonesia hanya terdiri sekitar 1% dari seluruh daratan di permukaan bumi, cadangan hutan alaminya merupakan yang terbesar di Asia dan kedua terbesar di dunia, yang diperkirakan membentang seluas lebih dari 100 juta hektar. Tak heran sumber daya alam di Indonesia amat melimpah. Sumber daya alam di hutan sangatlah beraneka ragam yang dapat dimanfaatkan manusia, salah satunya adalah kayu.

Kayu adalah bagian batang atau cabang serta ranting tumbuhan yang mengeras karena mengalami lignifikasi (pengayuan). Kayu digunakan untuk berbagai keperluan, mulai dari memasak, membuat perabot (meja, kursi), bahan bangunan (pintu, jendela, rangka atap), bahan kertas, dan banyak lagi. Kayu juga dapat dimanfaatkan sebagai hiasan-hiasan rumah tangga dan sebagainya.

Untuk pengolahan kayu hingga menjadi bahan jadi, pengrajin harus melakukan pemilihan jenis kayu sesuai dengan barang yang akan dibuat. Maka dari itu diperlukan penanganan yang berbeda untuk mengolah berbagai macam jenis kayu. Misalnya kayu Jati yang memiliki tekstur batang yang tergolong keras. Untuk jenis kayu yang keras memerlukan tenaga ekstra dan cenderung memakan waktu lama untuk memotong. Berbeda dengan kayu Albasia atau biasa disebut kayu Sengon yang memiliki tekstur lebih lunak.

Berdasarkan masalah di atas, maka dibuatlah sebuah pemotong kayu otomatis yang dapat membantu untuk mengoptimalkan hasil produksi dari segi pengolahan bahan dasar dan waktu pengerjaan. Pada laporan skripsi ini digunakan kontroler PID dalam pengaturan kecepatan motor pemotongnya, dengan harapan dapat mengatur kecepatan motor sesuai dengan yang diinginkan.

1.2. Rumusan Masalah

Berdasarkan uraian pada latar belakang, dapat disusun rumusan masalah sebagai berikut :

- 1) Bagaimana merancang dan membuat sistem kerja alat pemotong kayu yang dapat memotong dengan kecepatan yang relatif konstan meskipun tekstur kayunya yang berbeda-beda
- 2) Bagaimana merancang dan membuat suatu sistem pengaturan menggunakan kontroler PID (*Proporsional Integral Diferensial*) yang dapat menghasilkan respon sesuai dengan yang direncanakan.

1.3. Batasan Masalah

Dalam perancangan skripsi ini permasalahan dibatasi oleh hal-hal sebagai berikut :

- 1) Kerangka meja pemotong kayu berdimensi panjang 1 m, lebar 1 m, tinggi 40 cm.
- 2) Pengaturan yang dilakukan adalah pengaturan kecepatan motor berdaya 1050 W sesuai spesifikasi motor gergaji.
- 3) Pembahasan ditekankan pada penggunaan PID.
- 4) Kinerja *driver* dan perangkat elektronik tidak dibahas mendalam.
- 5) Kayu yang dipergunakan untuk pengujian hanya kayu Jati, kayu Mahoni, dan kayu Albasia.

1.4. Tujuan

Tujuan dari skripsi ini adalah untuk mempercepat proses pemotongan kayu menggunakan gergaji dengan metode kontrol PID.

1.5.Sistematika Pembahasan

Skripsi ini terdiri atas enam bab dengan sistematika pembahasan sebagai berikut:

BAB I Pendahuluan

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

BAB II Tinjauan Pustaka

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

BAB III Metodologi Penulisan

Membahas metode penelitian dan perencanaan alat.

BAB IV Perancangan dan Pembuatan Alat

Membahas perancangan kontroler PID (*Proporsional Deferensial Integral*) dan menerapkannya ke dalam *software*, sehingga sistem dapat bekerja dengan baik.

BAB V Pengujian dan Analisis

Membahas hasil pengujian sistem yang sudah dibuat, serta analisis hasil yang diperoleh.

BAB VI Penutup

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

BAB II

TINJAUAN PUSTAKA

Dalam merencanakan dan merealisasikan gergaji kayu ini, maka dibutuhkan pemahaman tentang berbagai hal yang mendukung sistem ini. Pemahaman ini akan bermanfaat untuk merancang perangkat keras dan perangkat lunak sistem. Pengetahuan yang mendukung perencanaan dan realisasi alat meliputi mesin Gergaji Listrik *Circular*, Motor, Sensor *Rotary encoder*, Mikrokontroler *Arduino Uno*, Kontrol PID.

2.1. Motor

Motor penggerak gergaji Hitachi Koki C7SS seperti pada Gambar 2.1 adalah gergaji bermata *Circular* atau melingkar yang dapat memotong berbagai macam jenis kayu. Dengan spesifikasi teknis yaitu:

- Voltase Listrik AC 220V
- *Input* daya terukur 1050W
- Kecepatan tanpa beban 5500 rpm
- Berat tanpa kabel 4,1 kg
- Diameter mata gergaji 190 mm
- Kedalaman potongan
 - Kedalaman potongan (90°) 68 mm
 - Kedalaman potongan (45°) 46 mm



Gambar 2.1 Hitachi Koki C7SS
Sumber: <http://www.hitachi.com/>

2.2. Sensor Optocoupler

Optocoupler adalah suatu piranti yang terdiri dari 2 bagian yaitu transmitter dan receiver, yaitu antara bagian cahaya dengan bagian deteksi sumber cahaya terpisah. Biasanya optocoupler digunakan sebagai saklar elektrik, yang bekerja secara otomatis. Optocoupler atau optoisolator merupakan komponen *coupling* (penggandeng) antara rangkaian input dengan rangkaian output yang menggunakan media cahaya (*opto*) sebagai penghubung. Dengan kata lain, tidak ada bagian yg konduktif antara kedua rangkaian tersebut. Optocoupler sendiri terdiri dari 2 bagian, yaitu transmitter (pengirim) dan *receiver* (penerima). Gambar 2.2 menunjukkan sensor optocoupler yang terpasang pada alat.



Gambar 2.2 Sensor Optocoupler

Transmitter merupakan bagian yg terhubung dengan rangkaian input atau rangkaian kontrol. Pada bagian ini terdapat sebuah LED infra merah (IR LED) yang berfungsi untuk mengirimkan sinyal kepada *receiver*. Padatransmitter dibangun dari sebuah LED infra merah. Jika dibandingkan dengan menggunakan LED biasa, LED infra merah memiliki ketahanan yang lebih baik terhadap sinyal tampak. Cahaya yang dipancarkan oleh LED infra merah tidak terlihat oleh mata telanjang.

Receiver merupakan bagian yg terhubung dengan rangkaian output atau rangkaian beban, dan berisi komponen penerima cahaya yang dipancarkan oleh *transmitter*. Komponen penerima cahaya ini dapat berupa photodiode ataupun phototransistor. Pada bagian *receiver* dibangun dengan dasar komponen *phototransistor*. *Phototransistor* merupakan suatu transistor yang peka terhadap tenaga cahaya. Suatu sumber cahaya

menghasilkan energi panas, begitu pula dengan spektrum infra merah. Karena spektrum infra mempunyai efek panas yang lebih besar dari cahaya tampak, maka *phototransistor* lebih peka untuk menangkap radiasi dari sinar infra merah.

Tujuan utama dari digunakan sensor cahaya dan piringan sensor adalah untuk mendapatkan data kecepatan putaran dari setiap roda. Piringan sensor yang digunakan dibuat dari negatif-film yang dijepit oleh dua buah *acrylic* transparan agar semakin presisi pembacaan datanya. Sedangkan sensor cahaya yang digunakan adalah optocoupler yang prinsip kerjanya adalah ketika ada benda yang berada di antara celah sensornya, maka cahaya yang dikirimkan tidak bisa diterima oleh bagian penerimanya, sehingga menghasilkan tegangan keluaran yang nilainya mendekati VCC, begitu juga sebaliknya, jika tidak ada benda diantara celah sensornya maka akan menghasilkan tegangan keluaran yang nilainya mendekati 0 Volt.

2.3. Kontroler

Sistem pengendalian dirancang untuk melakukan dan menyelesaikan tugas tertentu. Syarat utama sistem pengendalian adalah harus stabil. Di samping kestabilan mutlak, maka sistem harus memiliki kestabilan secara relatif, yakni tolak ukur kualitas kestabilan sistem dengan menganalisis sampai sejauh mana batas-batas kestabilan sistem tersebut jika dikenai gangguan (Ogata K.,1997). Selain itu analisis juga dilakukan untuk mengetahui bagaimana kecepatan sistem dalam merespons *input*, dan bagaimana peredaman terhadap adanya lonjakan (*over shoot*).

Suatu sistem dikatakan stabil jika diberi gangguan maka sistem tersebut akan kembali ke keadaan *steady state* di mana *output* berada dalam keadaan tetap seperti tidak ada gangguan. Sistem dikatakan tidak stabil jika *output*nya berosilasi terus menerus ketika dikenai suatu gangguan. Karena suatu sistem pengendalian biasanya melibatkan penyimpanan energi maka *output* sistem ketika diberi suatu *input*, tidak dapat mengikuti *input* secara serentak, tapi menunjukkan respon transien berupa suatu osilasi teredam sebelum mencapai *steady state*.

Dalam sistem pengendalian terdapat 2 macam *loop*:

- Pengendalian dengan *loop* terbuka
Sistem kontrol *loop* terbuka adalah sistem kontrol yang keluarannya tidak berpengaruh pada aksi pengontrolan. Jadi pada sistem kontrol *loop* terbuka, keluaran tidak diukur atau diumpan balik untuk dibandingkan dengan masukan.
- Pengendalian dengan *loop* tertutup
Sistem kontrol *loop* tertutup adalah sistem kontrol yang keluarannya mempunyai pengaruh langsung pada aksi pengontrolan. Disebut juga sistem kontrol yang menggunakan umpan balik untuk memperkecil kesalahan sistem.

2.3.1. Kontroler Proporsional

Kontroler proporsional memiliki keluaran yang sebanding dengan sinyal kesalahan (selisih antara besaran yang diinginkan dengan harga aktualnya). Untuk kontroler dengan aksi kontrol proporsional, hubungan antara keluaran kontroler $m(t)$ dan sinyal kesalahan penggerak $e(t)$ adalah:

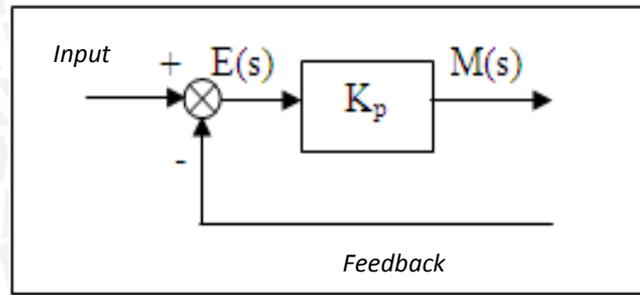
$$m(t) = K_p \cdot e(t) \quad (2-1)$$

atau, dalam besaran transformasi *Laplace*,

$$\frac{M(s)}{E(s)} = K_p \quad (2-2)$$

Di mana K_p adalah kepekaan proporsional atau penguatan.

Apapun wujud mekanisme yang sebenarnya dan apapun bentuk daya penggerakannya, kontroler proporsional pada dasarnya merupakan penguat dengan penguatan yang dapat diatur (Ogata K.,1997). Diagram blok kontroler proporsional ditunjukkan dalam Gambar 2.2



Gambar 2.3 Diagram Blok Kontroler Proporsional
 Sumber: Ogata K., 1997

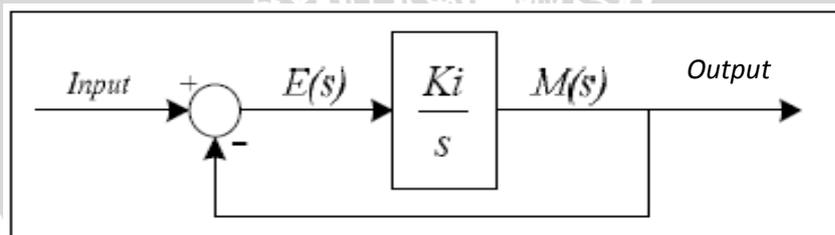
2.3.2. Kontroler Integral

Kontroler integral berfungsi mengurangi kesalahan keadaan mantap pada kontroler proporsional sebelumnya. Pada kontroler dengan aksi integral, harga keluaran kontroler $m(t)$ diubah dengan laju yang sebanding dengan sinyal kesalahan penggerak $e(t)$.

Jadi,

$$\frac{dm(t)}{dt} = K_i \cdot e(t) \tag{2-3}$$

Jika harga $e(t)$ diduakalikan, maka harga $m(t)$ berubah dengan laju perubahan menjadi dua kali semula. Jika kesalahan penggerak nol, maka harga $m(t)$ tetap stasioner. Aksi kontrol integral seringkali disebut control *reset* (Ogata K.,1997). Gambar 2.4 menunjukkan diagram blok kontroler integral.



Gambar 2.4 Diagram Blok Kontroler Integral
 Sumber: Ogata K., 1997

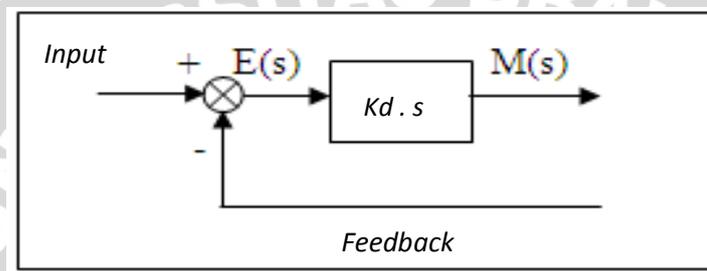
2.3.3. Kontroler Diferensial

Kontroler ini digunakan untuk memperbaiki atau mempercepat respons transien sebuah sistem kontrol dengan cara memperbesar *phase lead* terhadap penguatan kontrol

dan mengurangi *phase lag* penguatan tersebut (Ogata K.,1997). Kontroler diferensial tidak dapat mengeluarkan *output* bila tidak ada perubahan *input*, selain itu kontroler differensial tidak dapat digunakan untuk proses yang mengandung *noise*. Hubungan antara keluaran kontroler $m(t)$ dan sinyal kesalahan penggerak $e(t)$ adalah :

$$\frac{M(s)}{E(s)} = Kd \cdot s \tag{2-4}$$

Gambar 2.5 menunjukkan diagram blok kontroler diferensial.



Gambar 2.5 Diagram Blok Kontroler Diferensial
 Sumber: Teknik Kontrol Automatik, Katsuhiko Ogata, 1997

2.3.4. Kontroler Proporsional Integral Diferensial (PID)

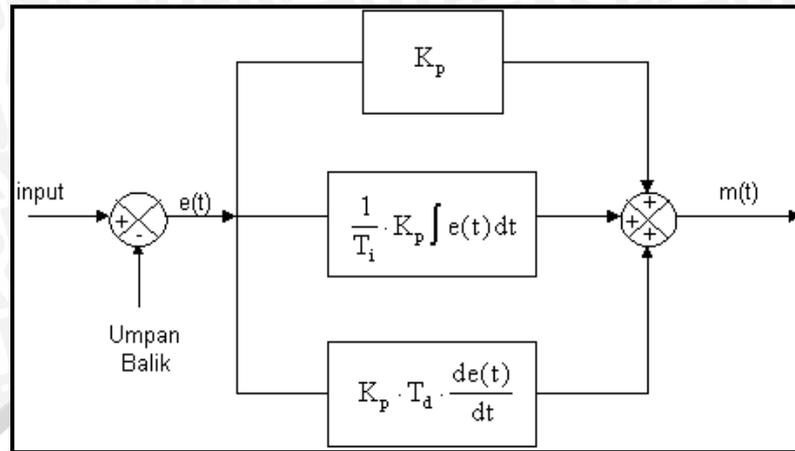
Gabungan aksi kontrol proporsional, integral, dan diferensial mempunyai keunggulan dapat saling menutupi kekurangan dan kelebihan dari masing-masing kontroler. Elemen-elemen P, I, dan D masing-masing secara keseluruhan bertujuan untuk mempercepat reaksi sebuah sistem, menghilangkan *offset* dan menghasilkan perubahan awal yang besar. Persamaan kontroler PID ini dapat dinyatakan sebagai berikut :

$$m(t) = Kp \cdot e(t) + \frac{Kp}{Ti} \cdot e(t)dt + Kp \cdot Td \frac{de(t)}{dt} \tag{2-5}$$

Dalam transformasi Laplace dinyatakan sebagai berikut :

$$\frac{M(s)}{E(s)} = Kp \left(1 + \frac{1}{Ti \cdot s} + Td \cdot s \right) \tag{2-6}$$

Gambar 2.6 menunjukkan diagram blok kontroler PID

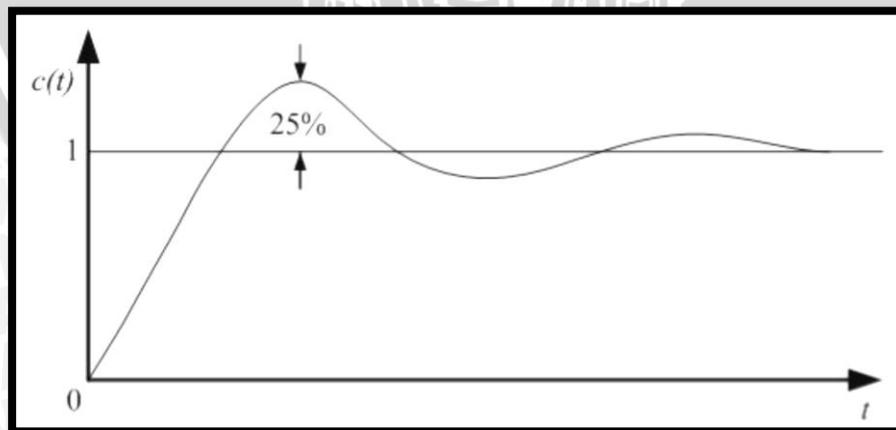


Gambar 2.6 Diagram Blok Kontroler PID
 Sumber: Ogata K., 1997

2.3.5. Metode Kontroler Proporsional Integral Diferensial (PID)

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 plan. (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, ditunjukkan dengan Gambar 2.7

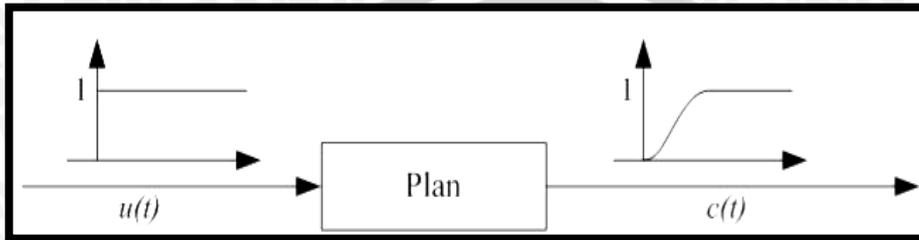


Gambar 2.7 Kurva Respon Unit Step yang Menunjukkan 25% Maximum Overshoot
 Sumber: Ogata, K., 1997



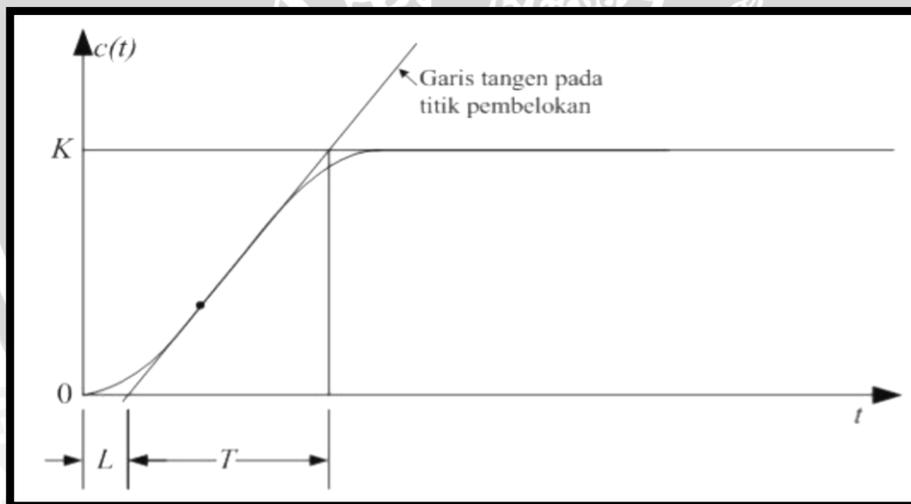
1). Metode Pertama

Metode pertama atau sering disebut metode kurva reaksi, respon dari plan dapat dapat diperoleh secara eksperimental dengan masukan berupa unit step, seperti yang ditunjukkan dengan Gambar 2.8



Gambar 2.8 Respon Plant Terhadap Masukan Berupa Unit Step
 Sumber: Ogata, K. 1997

Jika dalam plan tersebut terdapat integrator atau *dominan complex-conjugate poles*, maka kurva respon unit step berbentuk seperti huruf S, seperti ditunjukkan dengan Gambar 2.8 jika respon tidak memberikan bentuk kurva S, maka metode ini tidak berlaku.(Ogata, K., 1997).



Gambar 2.9 Kurva Respon yang Berbentuk S
 Sumber: Ogata, K. 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 Persamaan (2-7). Fungsi alih $C(s)/U(s)$ dapat dilakukan pendekatan dengan sistem orde satu dengan persamaan sebagai berikut:

$$\frac{c(s)}{U(s)} = \frac{Ke^{-Ls}}{Ts+1} \quad (2-7)$$

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., 1997).

Tabel 2.1 Aturan Penalaan Ziegler-Nichols Berdasarkan Respon Unit Step Dari Plan8

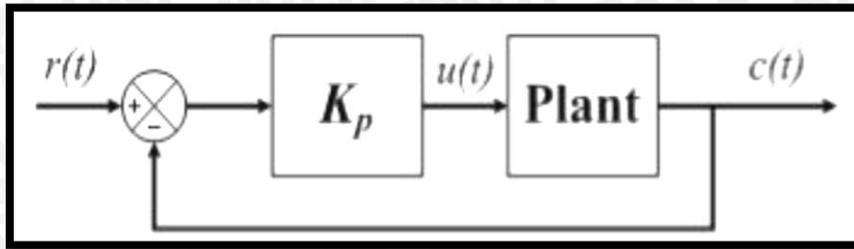
Tipe Kontroler	K_p	T_i	T_d
P	$\frac{T}{L}$	∞	0
PI	$0,9 \frac{T}{L}$	$\frac{L}{0,3}$	0
PID	$1,2 \frac{T}{L}$	$2L$	$0,5 L$

Sumber: Ogata, K. 1997

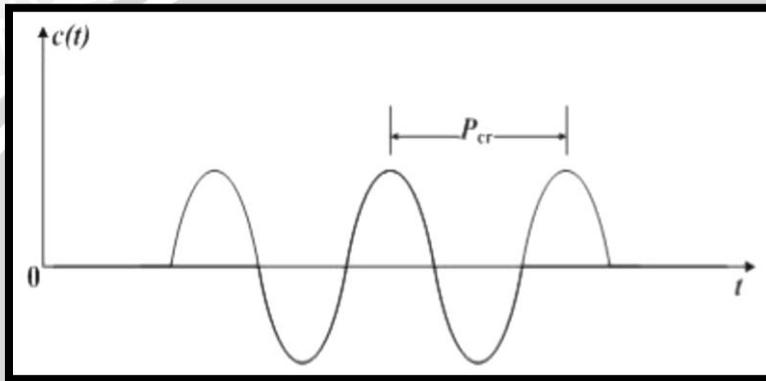
2). Metode Kedua

Dalam metode kedua ziegler-nichols, mula-mula yang dilakukan adalah membuat $T_i = \infty$ 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 ditunjukkan dengan Gambar 2.10 dan untuk osilasi berkesinambungan dengan periode P_{cr} ditunjukkan dengan Gambar 2.11



Gambar 2.10 Sistem Loop Tertutup dengan Kontroler Proporsional
 Sumber: Ogata, K., 1997



Gambar 2.11 Osilasi Berkesinambungan dengan Periode Pcr
 Sumber : Ogata, K., 1997

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).

Tabel 2.2 Aturan Dasar Ziegler-Nichols Berdasarkan Critical Gain Kcr dan Critical Period Pcr

Tipe Kontroler	K_p	T_i	T_d
P	$0.5 K_{cr}$	∞	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: Ogata, K., 1997



2.4. Komunikasi Serial RS232

Nama resmi dari *standart interface* ini adalah interface antara data *terminal equipment* dan data *communication employing serial binary data interchange*, yaitu suatu terminal yang menghubungkan antara terminal data dari suatu peralatan dan peralatan komunikasi data yang menjalankan pertukaran data *biner* secara serial, oleh industri komunikasi data disebut *RS-232*. Sejak dipublikasikan pertama kali pada tahun 1962, *interface* ini telah mengalami beberapa revisi dan sekarang yang banyak dipergunakan adalah *RS-232* revisi kelima yang dipergunakan pada tahun 1991 dan disebut *RS-232E* atau *EIA-232-E*. Awalan “RS” pada *RS-232* merupakan singkatan dari *recommended standart* yang berarti *standart* yang dianjurkan, karena selama ini dalam publikasi *EIA* tidak pernah memiliki ketetapan hukum yang mengharuskan untuk digunakan dalam dunia komunikasi data. Ada beberapa standar sederhana yang dikembangkan sendiri oleh pembuat peralatan elektronik yang mungkin dapat diikuti, agar secara umum ada kecocokan. Standar lain yang menyangkut referensi aspek fungsi dan prosedur *interface* secara mekanikal dan elektrik yang dipublikasikan oleh *ITU-T* pada tahun 1993 adalah *standart V.25* dan *V.28*, sehingga *RS-232E* sebenarnya mencakup empat aspek, yaitu Mechanical (*ISO21110*), Electrical (*V.28*), Functional (*V.24*), Prosedural (*V.24*).

Terdapat beberapa macam cara untuk menerapkan *interface* data biner pada komunikasi secara serial, salah satunya adalah *RS-232* yang merupakan salah satu dari standart yang dipilih dan sekarang telah dipakai secara luas dan dalam komunikasi data umumnya digunakan untuk menghubungkan *DTE (Data Terminal Equipment)* ke *DCE (Data Communication Equipment)* yang berupa peralatan sistem komunikasi analog.

Protokol standar yang mengatur komunikasi melalui serial port disebut *RS-232 (Recommended Standard-232)* yang dikembangkan oleh *EIA (Electronic Industries Association)*. Interfacing *RS-232* menggunakan komunikasi asynchronous di mana sinyal clock tidak dikirimkan bersamaan dengan data. Setiap *word* data disinkronisasikan menggunakan sebuah *start bit* dan sebuah *stop bit*. Jadi, sebuah *frame* data terdiri dari sebuah *start bit*, diikuti bit-bit data dan diakhiri dengan *stop bit*. Jumlah bit data yang digunakan dalam komunikasi serial adalah 8 bit. *Encoding* yang digunakan dalam

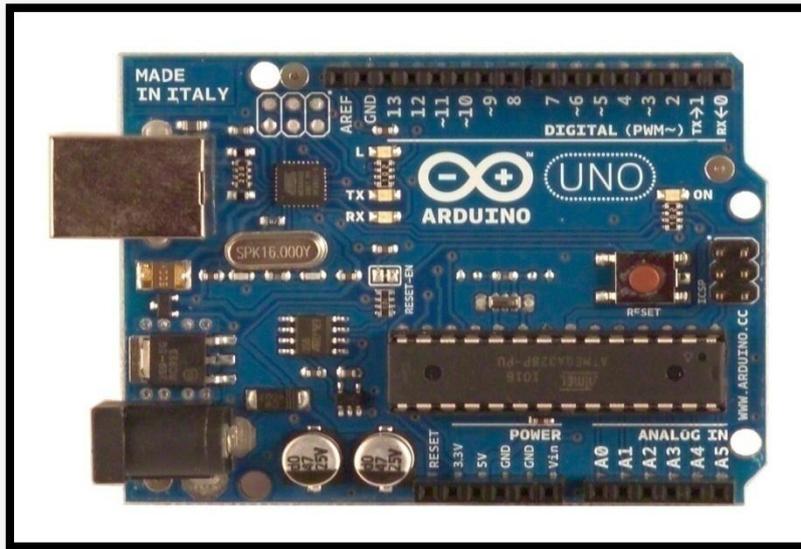
komunikasi serial adalah NRZ (*Non-Return-to-Zero*), di mana bit 1 dikirimkan sebagai *high value* dan bit 0 dikirimkan sebagai *low value*.

Dalam *interfacing RS-232*, tegangan negatif merepresentasikan bit 1 dan tegangan positif merepresentasikan bit 0. *RS232 Serial port* juga merupakan rangkaian *converter* komunikasi antara mikrokontroler ke PC atau sebaliknya. Mikrokontroler mempunyai *level TTL low logic* 0 – 1.8V dan *TTL high logic* 2.2V – 5V, sedangkan untuk PC memiliki *high logic* 5V – 12V dan untuk *low logic* -5V – (-12V), sehingga diperlukan *converter*. Komunikasi *RS232* berupa komunikasi asinkron dengan baud rate 9600 bps, 8 bit data, *parity none* dan *start* bit 1.

2.5. Arduino Uno

Arduino Uno adalah *board* mikrokontroler berbasis *ATmega328*. Memiliki 14 pin input dari output digital dimana 6 pin input tersebut dapat digunakan sebagai output PWM dan 6 pin *input analog*, 16 MHz osilator kristal, koneksi USB, *jack power*, ICSP header, dan tombol reset. Untuk mendukung mikrokontroler agar dapat digunakan, cukup hanya menghubungkan Board *Arduino Uno* ke komputer dengan menggunakan kabel USB atau listrik dengan AC yang-ke adaptor-DC atau baterai untuk menjalankannya.

Arduino Uno berbeda dengan semua board sebelumnya dalam hal koneksi *USB-to-serial* yaitu menggunakan fitur *Atmega8U2* yang diprogram sebagai konverter *USB-to-serial* berbeda dengan board sebelumnya yang menggunakan chip *FTDI driver USB-to-serial*.



Gambar 2.12. Tampak depan Arduino Uno

Nama “Uno” berarti satu dalam bahasa Italia, untuk menandai peluncuran *Arduino* 1.0. Uno dan versi 1.0 akan menjadi versi referensi dari *Arduino*. Uno adalah yang terbaru dalam serangkaian *board* USB *Arduino*, dan sebagai model referensi untuk *platform* *Arduino*, untuk perbandingan dengan versi sebelumnya,

2.5.1. Daya

Arduino Uno dapat diaktifkan melalui koneksi USB atau dengan catu daya eksternal (otomatis). Eksternal (non-USB) daya dapat berasal baik dari AC-ke adaptor-DC atau baterai. Adaptor ini dapat dihubungkan dengan menancapkan *plug jack* pusat-positif ukuran 2.1mm konektor *POWER*. Ujung kepala dari baterai dapat dimasukkan kedalam Gnd dan Vin pin *header* dari konektor *POWER*. Kisaran kebutuhan daya yang disarankan untuk board Uno adalah 7 sampai dengan 12 volt, jika diberi daya kurang dari 7 volt kemungkinan pin 5v Uno dapat beroperasi tetapi tidak stabil kemudian jika diberi daya lebih dari 12V, regulator tegangan bisa panas dan dapat merusak *board* Uno.

Pin listrik adalah sebagai berikut:

VIN. Tegangan masukan kepada *board Arduino* ketika itu menggunakan sumber daya eksternal (sebagai pengganti dari 5V koneksi USB atau sumber daya lainnya) 5V. Catu daya digunakan untuk daya mikrokontroler dan komponen lainnya. Sebuah pasokan 3,3 volt dihasilkan oleh *regulator on-board*.

GND. *Ground pin*.

2.5.2. Memori

ATmega328 memiliki 32 KB (dengan 0,5 KB digunakan untuk bootloader), 2 KB dari SRAM dan 1 KB EEPROM

2.5.3. Input dan Output

Masing-masing dari 14 pin digital di *Uno* dapat digunakan sebagai input atau output, dengan menggunakan fungsi *pinMode* (), *digitalWrite* (), dan *digitalRead* (), beroperasi dengan daya 5 volt. Setiap pin dapat memberikan atau menerima maksimum 40 mA dan memiliki *internal pull-up* resistor (secara default terputus) dari 20-50 kOhms. Selain itu, beberapa pin memiliki fungsi khusus:

Serial: 0 (RX) dan 1 (TX). Digunakan untuk menerima (RX) dan mengirimkan (TX) TTL data serial. Pin ini dihubungkan ke pin yang berkaitan dengan chip Serial ATmega8U2 USB-to-TTL.

Eksternal menyela: 2 dan 3. Pin ini dapat dikonfigurasi untuk memicu *interrupt* pada nilai yang rendah, dengan batasan tepi naik atau turun, atau perubahan nilai.

PWM: 3, 5, 6, 9, 10, dan 11. Menyediakan output PWM 8-bit dengan fungsi *analogWrite* () .

SPI: 10 (SS), 11 (Mosi), 12 (MISO), 13 (SCK). Pin ini mendukung komunikasi SPI menggunakan *SPI library*.

LED: 13. Ada built-in *LED* terhubung ke pin digital 13. Ketika pin bernilai nilai *HIGH*, *LED on*, ketika pin bernilai *LOW*, *LED off*.

Uno memiliki 6 masukan *analog*, berlabel A0 sampai dengan A5, yang masing-masing menyediakan 10 bit dengan resolusi (yaitu 1024 nilai yang berbeda). Selain itu, beberapa pin memiliki fungsi khusus:

I2C: A4 (*SDA*) dan A5 (*SCL*). Dukungan *I2C* (*TWI*) komunikasi menggunakan perpustakaan *Wire*.

Aref. Tegangan referensi (0 sampai 5V saja) untuk *input analog*. Digunakan dengan fungsi *analogReference* ().

Reset. Bawa baris ini *LOW* untuk me-*reset* mikrokontroler.

2.5.4. Komunikasi

Arduino Uno memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, *Arduino* lain, atau mikrokontroler lainnya. ATmega328 menyediakan UART TTL (5V) untuk komunikasi serial, yang tersedia di pin digital 0 (RX) dan 1 (TX). Sebuah ATmega8U2 sebagai saluran komunikasi *serial* melalui USB dan sebagai *port virtual com* untuk perangkat lunak pada komputer. *Firmware* '8 U2 menggunakan *driver* USB standar COM, dan tidak ada *driver* eksternal yang diperlukan. Namun, pada *Windows* diperlukan, sebuah file inf. Perangkat lunak *Arduino* terdapat monitor serial yang memungkinkan digunakan memonitor data tekstual sederhana yang akan dikirim ke atau dari board *Arduino*. LED RX dan TX di papan tulis akan berkedip ketika data sedang dikirim melalui *chip USB-to-serial* dengan koneksi USB ke komputer (tetapi tidak untuk komunikasi serial pada pin 0 dan 1).

Sebuah *Software Serial library* memungkinkan untuk berkomunikasi secara serial pada salah satu pin digital pada board *Uno*'s. ATmega328 juga mendukung *I2C* (*TWI*) dan komunikasi *SPI*. Perangkat lunak *Arduino* termasuk perpustakaan kawat untuk menyederhanakan penggunaan bus *I2C*.

BAB III

METODOLOGI PENELITIAN

Untuk menyelesaikan rumusan masalah dan merealisasikan tujuan penelitian yang terdapat di bab pendahuluan maka diperlukan metode untuk menyelesaikan masalah tersebut.

3.1. Studi Literatur

Pengetahuan yang mendukung perencanaan dan realisasi alat meliputi:

- 1) Gergaji Hitachi Koki C7SS
- 2) Sensor Optocoupler
- 3) Kontrol PID
- 4) Mikrokontroler *Arduino Uno*

3.2. Perancangan dan Pembuatan Alat

3.2.1. Perancangan dan Pembuatan Perangkat Keras (*Hardware*)

- 1) Pembuatan blok diagram lengkap sistem keseluruhan
- 2) Desain *Printed Circuit Board* (PCB) menggunakan *software Eagle Layout Editor*
- 3) Merakit perangkat keras masing-masing blok

3.2.2. Perancangan dan Pembuatan Perangkat Lunak (*Software*)

Setelah kita mengetahui seperti apa perangkat keras yang dirancang, maka kita membutuhkan perangkat lunak untuk mengatur kerja dari alat ini. Desain dan parameter yang telah dirancang kemudian diterapkan kedalam mikrokontroler *Arduino Uno*.

3.3. Pengujian Alat

Untuk memastikan bahwa sistem ini berjalan sesuai yang direncanakan maka perlu dilakukan pengujian alat meliputi perangkat keras (*hardware*) yang dilakukan baik tiap blok rangkaian maupun keseluruhan sistem.

3.3.1. 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.

3.3.2. Pengujian Keseluruhan Sistem

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

3.4. Pengambilan Kesimpulan

Pengambilan kesimpulan dilakukan setelah didapatkan hasil dari pengujian. Jika hasil yang diperoleh telah sesuai dengan spesifikasi yang direncanakan maka alat tersebut telah memenuhi harapan dan memerlukan pengembangan untuk penyempurnaannya.

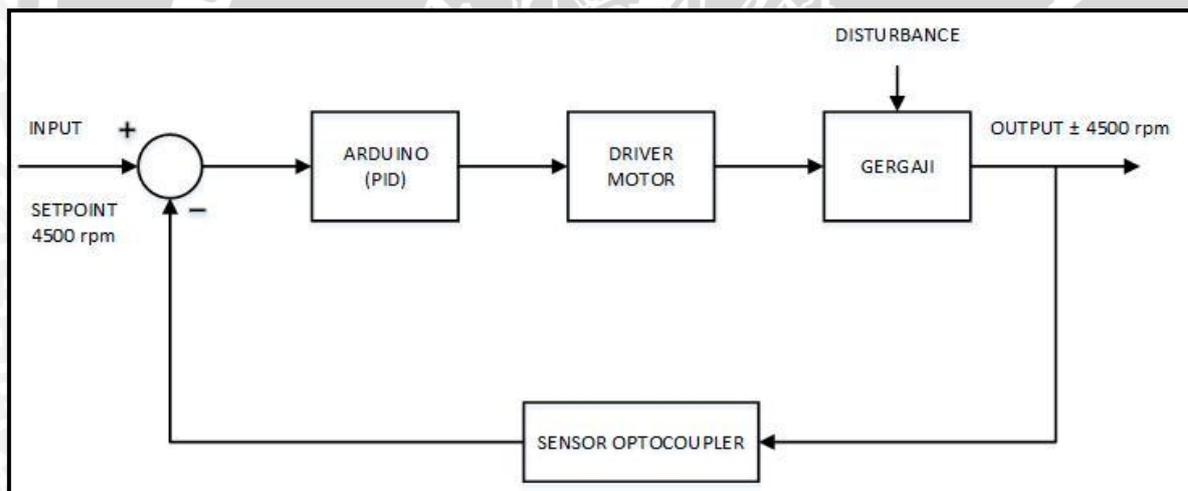
BAB IV

PERANCANGAN & PEMBUATAN ALAT

Pada bab ini akan dijelaskan mengenai perancangan pemotong kayu mulai dari diagram blok sistem, perancangan perangkat keras, dan perancangan perangkat lunak. Perancangan dan pembuatan dilakukan secara bertahap dan sistematis, sehingga nantinya akan memudahkan dalam analisis sistem.

4.1. Diagram Blok Sistem

Pada perancangan alat diperlukan perancangan blok diagram sistem yang dapat menjelaskan sistem secara garis besar dan diharapkan alat dapat bekerja sesuai dengan rencana. Blok diagram tersebut dapat dilihat dari Gambar 4.1 berikut.



Gambar 4.1 Diagram Blok Sistem Pengaturan

Diagram blok tersebut menjelaskan diagram blok sistem pengaturan pengolahan data dari sensor hingga ke aktuatornya (motor gergaji Hitachi Koki C7SS). *Input* berupa data yang dibaca sensor *optocoupler* yang mendeteksi kecepatan putar motor gergaji. Data keluaran sensor pada awal pembacaan adalah kecepatan *starting* motor saat pertama

dinyalakan sekaligus sebagai input pada kontroler *Proportional Integral Diferensial* (PID). *Output* PID berupa perintah pada motor agar menyesuaikan kecepatan sesuai dengan *set point* yang diharapkan. Apabila *output* belum sesuai dengan *set point* yang diharapkan maka akan diproses lagi oleh kontroler. Namun apabila *output* telah sesuai dengan *set point* yang diharapkan maka proses dinyatakan selesai. *Output* dari keseluruhan sistem ini adalah kecepatan yang hasilnya mendekati *set point*.

4.2. Prinsip Kerja Alat

Alat ini bekerja dengan adanya masukan dari sensor *optocoupler* yang di kopel dengan gergaji. Sinyal dari *optocoupler* akan menjadi masukan mikrokontroler yang akan mengolah data. Data dari mikrokontroler menjadi masukan ke *driver* motor untuk memutar motor gergaji agar mempertahankan kecepatan motor sesuai dengan yang diinginkan. Kecepatan motor akan dipertahankan selama waktu yang ditentukan hingga proses pemotongan berhenti.

4.3. Perancangan Perangkat Keras

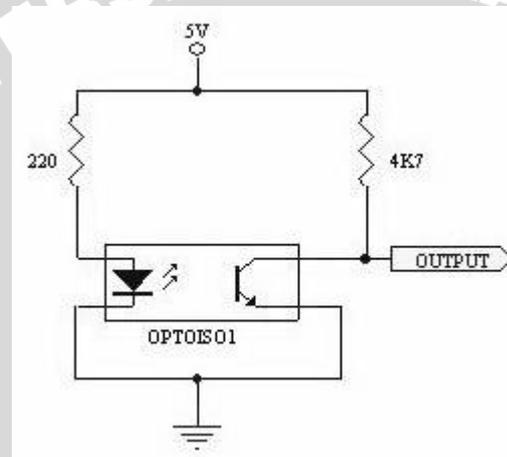
4.3.1. Perancangan Mekanik

Perancangan mekanik dijelaskan sebagai berikut:

1. Gergaji secara keseluruhan memiliki dimensi panjang 28 cm, lebar 15 cm, dan tinggi 20 cm.
2. Gergaji dipasang pada meja dengan kerangka besi berdimensi panjang 1 m, lebar 1 m, dan tinggi 1m.
3. Sebuah motor DC yang hubungkan dengan arduino untuk dikendalikan kecepatannya dengan merubah tegangannya.
4. Sebuah motor DC sebagai penggerak gergaji agar bisa maju dan mundur.
5. Sebuah sensor *optocoupler* sebagai sensor kecepatan yang dikopel pada gergaji.
6. Sebuah Arduino Uno sebagai kontroler.
7. Catu tegangan dengan sumber AC 220V.

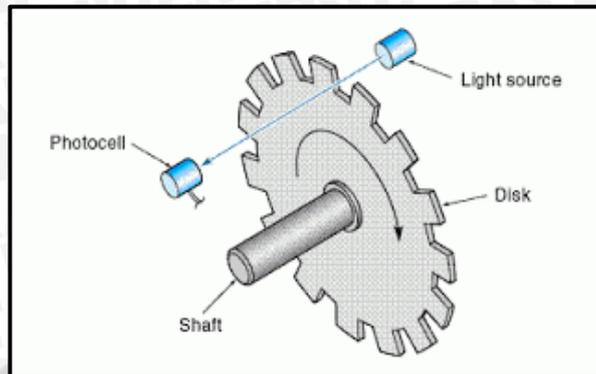
4.3.2. Perancangan Sensor Optocoupler

Optocoupler memiliki karakteristik penerima (photo transistor) dan akan mengalami perubahan logika bila terjadi perubahan intensitas cahaya yang dipancarkan oleh pemancar (LED infra merah) untuk penerima. Kecerahan led berbanding lurus dengan arus diodanya. Karena arus kolektor sebanding dengan tingkat kecerahan dari led maka dapat dikatakan bahwa arus dioda mengendalikan arus kolektor seperti transistor pada umumnya. Biasanya arus yang diperbolehkan mengalir pada infra merah adalah berkisar pada 15 hingga 25 miliampere. Rangkaian sensor optocoupler dapat dilihat pada Gambar 4.2



Gambar 4.2 Rangkaian Sensor Optocoupler
Sumber: Elektronika-dasar.web.id

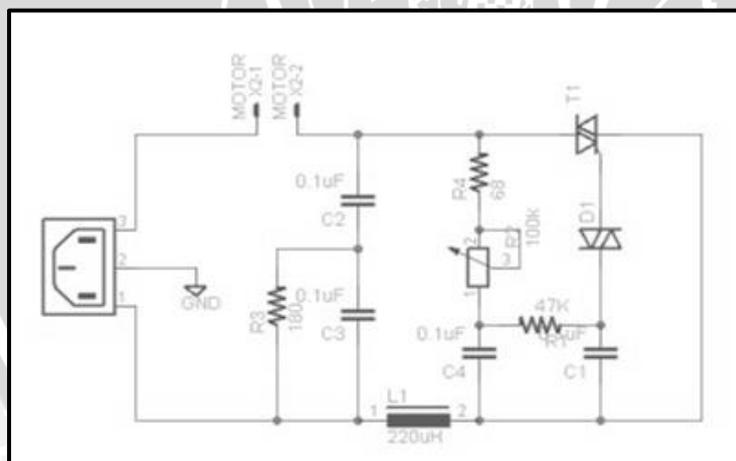
Photocell merupakan jenis transistor yang peka terhadap cahaya infra merah. Piringan yang terletak di antara Photocell dan Light Source (LED) tersebut berfungsi untuk mempengaruhi intensitas cahaya yang diberikan oleh LED ke *photocell* yang akan memberikan perubahan level logika sesuai dengan putaran piringan. Kecepatan perubahan logika *photocell* akan sebanding dengan kecepatan putaran roda cacah. Desain umum Optocoupler dapat dilihat pada Gambar 4.3



Gambar 4.3 Desain Umum Optocoupler
 Sumber: <http://ezkhel-design.blogspot.co.id>

4.3.3. Perancangan Rangkaian Pengatur Kecepatan Motor Gergaji

Gambar 4.4 menunjukkan rangkaian pengatur kecepatan motor gergaji. Pengaturan kecepatan dilakukan dengan cara mengubah tegangannya. Rangkaian Pengatur Kecepatan Motor Gergaji pada gambar 4.4 bekerja secara *analog* dengan cara mengatur *supply* tegangan motor gergaji menggunakan TRIAC berdasarkan tegangan bias yang diberikan ke TRIAC tersebut melalui DIAC yang dikendalikan oleh potensiometer R1.



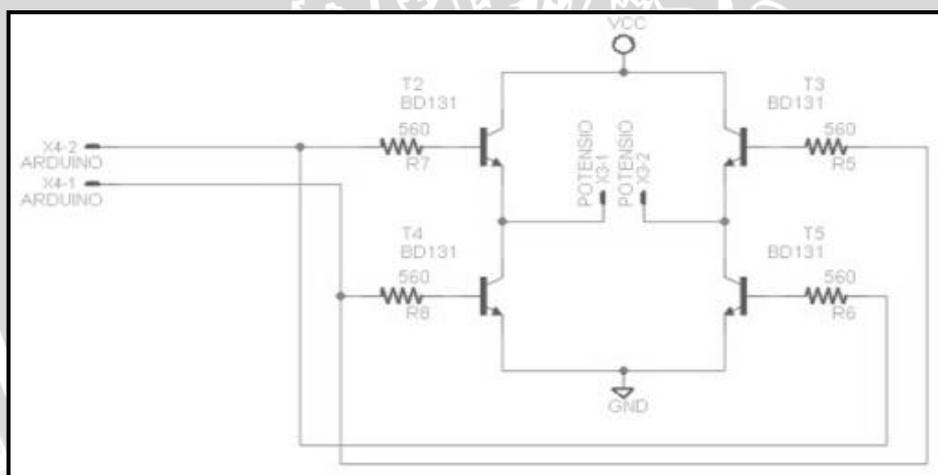
Gambar 4.4 Rangkaian Pengaturan Kecepatan Motor Gergaji

Gambar 4.4 menunjukkan rangkaian pengatur kecepatan motor gergaji. Pengaturan kecepatan dilakukan dengan cara mengubah tegangannya. Rangkaian Pengatur Kecepatan Motor Gergaji pada gambar 4.4 bekerja secara *analog* dengan cara mengatur *supply*

tegangan motor gergaji menggunakan TRIAC berdasarkan tegangan bias yang diberikan ke TRIAC tersebut melalui DIAC yang dikendalikan oleh potensiometer R1. Dimana semakin besar bias gate TRIAC yang diberikan maka semakin besar pula level tegangan yang diberikan ke motor Gergaji, dan semakin kecil tegangan bias gate maka semakin rendah juga tegangan *supply* ke motor gergaji. Semakin besar tegangan *supply* ke motor gergaji yang diberikan maka semakin cepat pula kecepatan putaran motor gergaji dan sebaliknya akan semakin lambat putaran motor apabila *supply* tegangan ke motor gergaji diturunkan.

4.3.4. Pembuatan Rangkaian Pengatur Arah Putaran Motor DC

Gambar 4.5 menunjukkan rangkaian pengatur arah putaran motor penggerak potensio. Banyak cara untuk mengatur kecepatan motor, karena kecepatan motor berbanding lurus dengan potensial yang diberikan maka kita dapat mengatur kecepatan motor dengan mengecilkan dan membesarkan potensial tegangan pada motor.



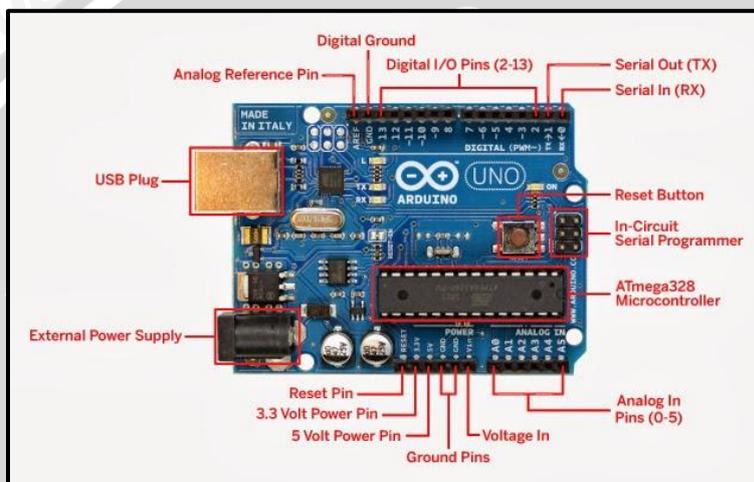
Gambar 4.5 Schematic Pengaturan Arah Putaran Motor Penggerak Potensio

Cara lain adalah dengan menggunakan teknik PWM (Pulse Width Modulation). Teknik ini paling sering digunakan untuk kontrol motor pada aktuator robot. Teknik ini mengatur pengaturan lebar pulsa yang diberikan kepada motor. Semakin besar (lebar pulsa) maka kecepatan motor akan menuju maksimal. Pengendalian kecepatan motor dalam

perancangan ini yaitu dengan merubah besar kecilnya potensial tegangan pada motor melalui program. Jadi pengendaliannya dilakukan secara otomatis.

4.3.5. Konfigurasi Arduino Uno

Pada alat ini digunakan mikrokontroler Arduino. Arduino menerima data berupa kecepatan yang dihasilkan gergaji. Kemudian data tersebut diproses dengan menggunakan PWM motor. Konfigurasi Arduino Uno dapat dilihat pada Gambar 4.6



Gambar 4.6 Konfigurasi microcontroller Arduino Uno
 Sumber <http://thietbichetao.com>

Tabel 4.1 Deskripsi konfigurasi Arduino Uno dengan sensordan motor DC

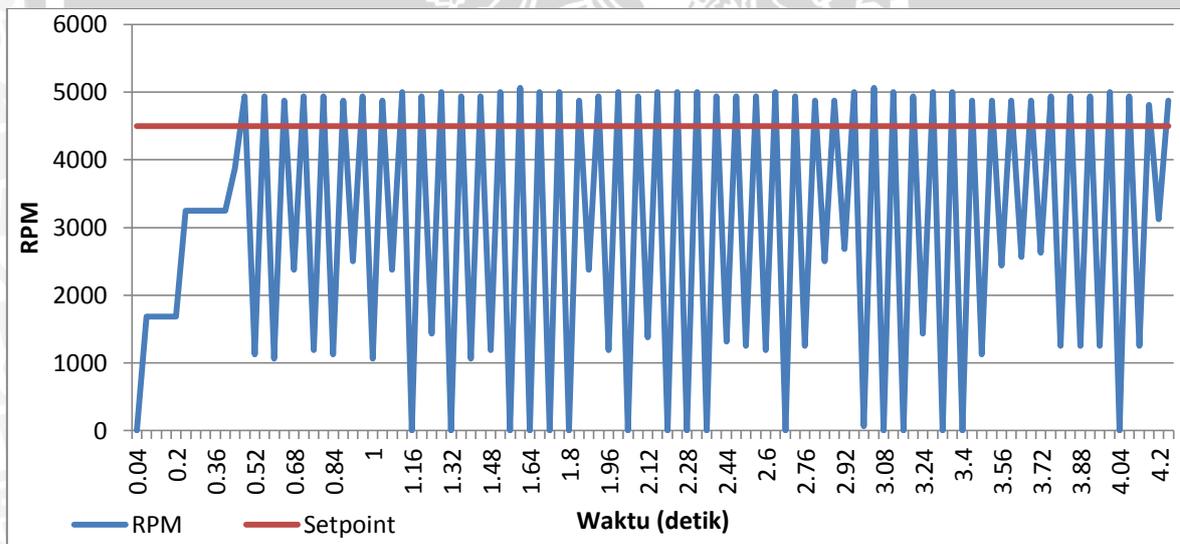
Nama	Kaki Arduino	Deskripsi
Sensor	PWM 2	Input (I/O digital)
	GND	Power
	5V	
Speed Controller (Driver Motor)	PWM 5 & 6	Output (I/O digital)
	A0	Analog in
	Vin	Power
	GND	
	5V	

Dari Tabel 4.1 dapat dijelaskan deskripsi konfigurasi Arduino Uno dengan sensor dan *speed controller* yang mana sensor *optocoupler* dikonfigurasi pada kaki arduino PWM 2. Kemudian dari hasil pembacaan sensor diproses oleh kontroler apabila telah sesuai dengan set point maka *speed controller* yang di konfigurasi pada kaki arduino PWM 5 dan PWM 6 otomatis mengatur potensial tegangan sehingga *output* yang berupa kecepatan gergaji dapat dikendalikan sesuai set point yang diinginkan.

4.4. Perancangan Perangkat Lunak

4.4.1. Perancangan Kontroler PID

Perancangan kontroler dilakukan dengan menggunakan *serial monitor* program Arduino Uno untuk mengamati respon kecepatan motor gergaji. Langkah pertama adalah dengan melihat respon kecepatan motor setelah diberi kenaikan $K_c = 3.6$ seperti dalam Gambar 4.7.



Gambar 4.7 Respon PID Zyglor Nichols

Berdasarkan Gambar 4.7 didapatkan nilai P_{cr} sebesar 4. Nilai parameter kontroler PID ditentukan berdasarkan tabel aturan dasar Ziegler-Nichols dengan *critical gain* K_{cr} dan *critical period* P_{cr} yang ditunjukkan dalam Tabel 4.2

Tabel 4.2 Dasar Ziegler-Nichols Berdasarkan *Critical Gain* K_{cr} dan *Critical Period* P_{cr} (Ogata K., 1997)

Tipe Kontroler	K_p	T_i	T_d
P	$0.5 K_{cr}$	∞	0
PI	$0.45 K_{cr}$	$\frac{1}{1.2} P_{cr}$	0
PID	$0.6 K_{cr}$	$0.5 P_{cr}$	$0.125 P_{cr}$

maka nilai parameter PID yang diperoleh adalah :

$$K_p = 0.6 \times 3.6 = 2.16$$

$$T_i = 0.5 \times 4 = 2$$

$$T_d = 0.125 \times 4 = 0.5$$

Dengan demikian dapat ditentukan nilai K_p , K_i , dan K_d adalah:

$$K_p = 2.16$$

$$K_i = \frac{K_p}{T_i}$$

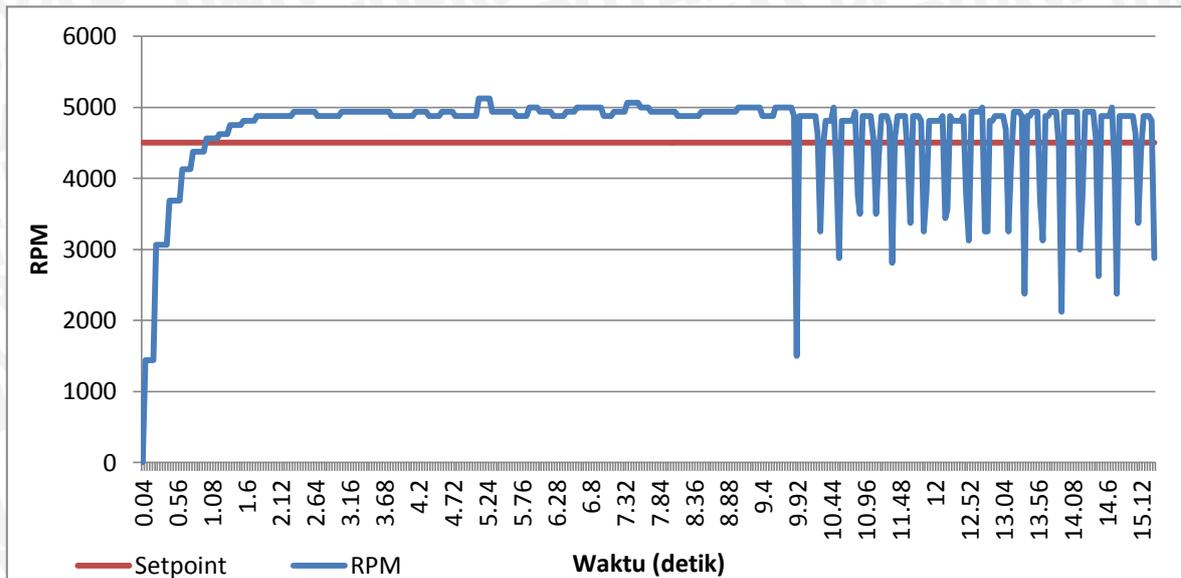
$$= 2.16 / 2$$

$$= 1.08$$

$$K_d = K_p \times T_d$$

$$= 2.16 \times 0.5 = 1.08$$

Dari penghitungan penentuan nilai penguatan dari metode kedua Ziegler-Nichols di atas diperoleh $K_p = 2.16$, $K_i = 1.08$, $K_d = 1.08$ yang akan digunakan untuk pengendali kecepatan motor. Dengan nilai K_p , K_i , dan K_d yang sudah diperoleh, maka grafik sistem respon yang diperoleh ditunjukkan oleh Gambar 4.8



Gambar 4.8 Respon Sistem dengan K_p , K_i , dan K_d Berdasarkan Ziegler-Nichols

Dari grafik Gambar 4.8 bisa dilihat sistem tidak dapat menggunakan perhitungan Ziegler-Nichols karena respon yang dihasilkan terus mengalami osilasi dan tidak bisa *steady* dan *overshoot* terlalu besar. Maka dilakukan *hand tuning* (*tuning* eksperimen) untuk mendapatkan respon yang lebih baik.

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 sistem pengendalian kecepatan ini, menggunakan metode *handtuning* untuk menentukan parameter K_p , K_i , dan K_d .

Menurut Smith (1979), untuk melakukan *hand tuning* prosedur yang dilakukan adalah sebagai berikut :

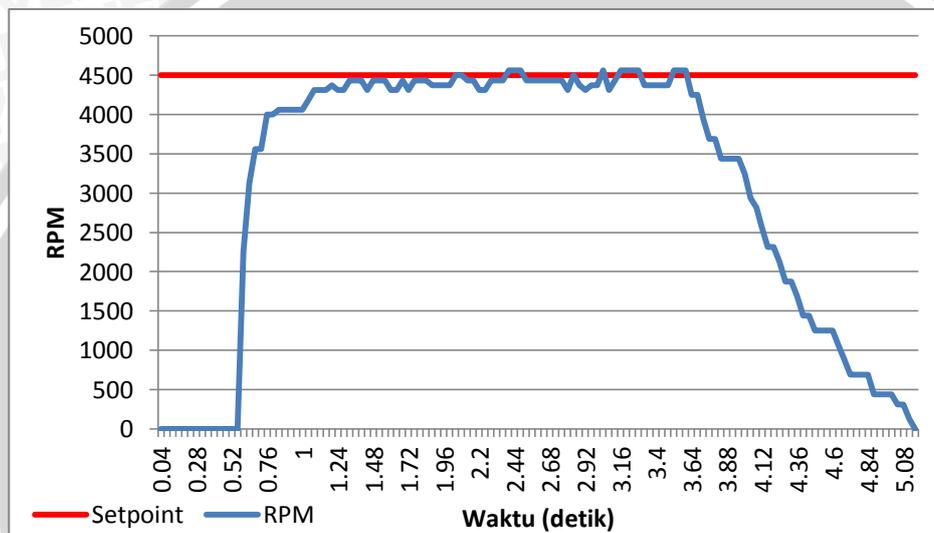
1. Melepaskan kontroler integral dan diferensial dengan memberikan nilai $K_i=0$ dan $K_d=0$.
2. Mengatur nilai K_p hingga didapatkan respon yang diinginkan, dengan mengabaikan *offset* dari *setpoint*.
3. Dengan terus menaikkan nilai K_p nilai dari K_d dinaikkan untuk mengurangi *overshoot* yang terjadi.

4. Naikkan nilai K_i untuk mengurangi *offset*.

Dengan menggunakan metode *hand tuning* nilai parameter PID perlu diubah-ubah secara *trial* dan *error* agar respon yang diperoleh sesuai dengan harapan.

Proses *tuning* parameter PID ini dilakukan dengan cara mengatur nilai K_p hingga didapatkan respon sistem yang mendekati *setpoint* 5000 rpm.

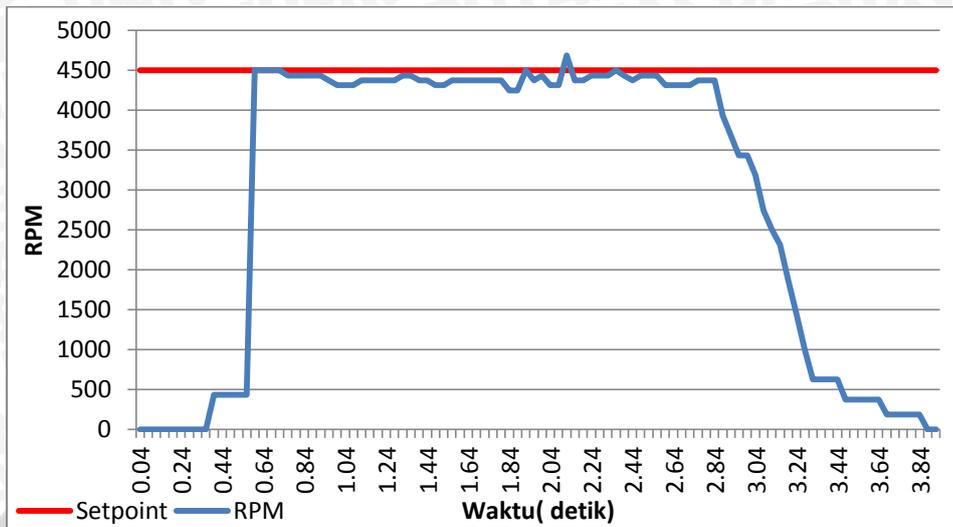
Untuk nilai $K_p = 0.1$, $K_i = 0$, dan $K_d = 0$ diperoleh grafik respon seperti pada Gambar 4.9



Gambar 4.9 Grafik Respon untuk $K_p = 0.1$

$$\begin{aligned} \% E_{ss} &= \frac{|\text{Averagespeedsteady} - \text{setpoint}|}{\text{setpoint}} \times 100\% \\ &= \frac{|4421.5 - 4500|}{4500} \times 100\% \\ &= 1.744\% \text{ dan memiliki waktu } \textit{steady} \text{ 2.32 detik.} \end{aligned}$$

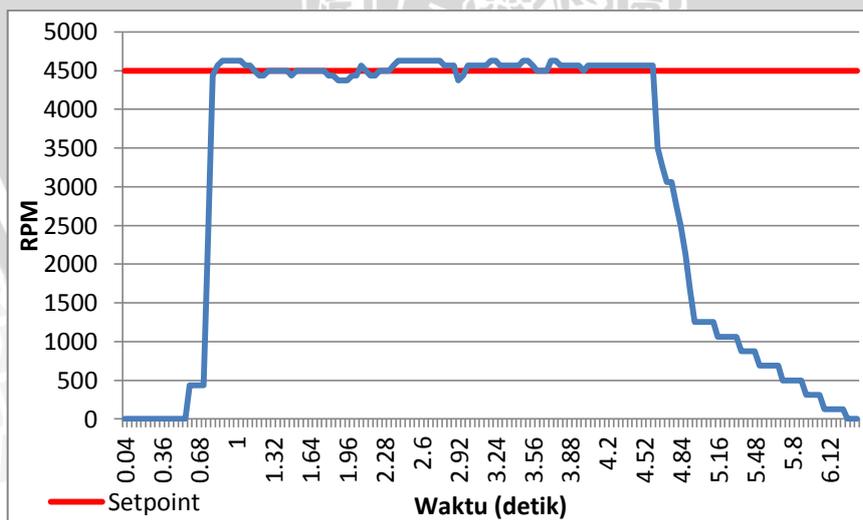
Untuk nilai $K_p = 0.3$, $K_i = 0$, dan $K_d = 0$ diperoleh grafik respon seperti pada Gambar 4.10



Gambar 4.10 Grafik Respon untuk $K_p = 0.3$

$$\begin{aligned} \% E_{ss} &= \frac{|Average\ speed\ steady - setpoint|}{setpoint} \times 100\% \\ &= \frac{|4384727 - 4500|}{4500} \times 100\% \\ &= 2.562\% \text{ dan memiliki waktu } steady \text{ 2.04 detik.} \end{aligned}$$

Untuk nilai $K_p = 0.4$, $K_i = 0$, dan $K_d = 0$ diperoleh grafik respon seperti pada Gambar 4.11



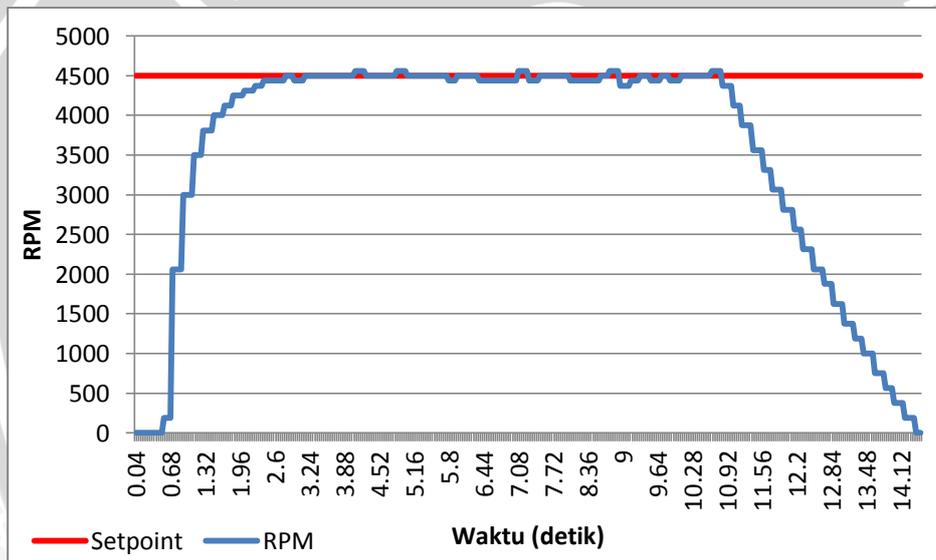
Gambar 4.11 Grafik Respon untuk $K_p = 0.4$



$$\begin{aligned} \% E_{ss} &= \frac{|Average\ speed\ steady - setpoint|}{setpoint} \times 100 \% \\ &= \frac{|4537.235 - 4500|}{4500} \times 100 \% \\ &= 0.8275\% \text{ dan memiliki waktu } steady \text{ 1.28 detik.} \end{aligned}$$

Dari hasil perancangan tersebut, didapatkan nilai $K_p = 0.4$ dan dapat mencapai stabil serta memiliki *error steady state* paling kecil walaupun masih terjadi. Setelah diperoleh nilai K_p yang cukup baik, Untuk mengurangi *overshoot* maka digunakanlah kontroler *integral*.

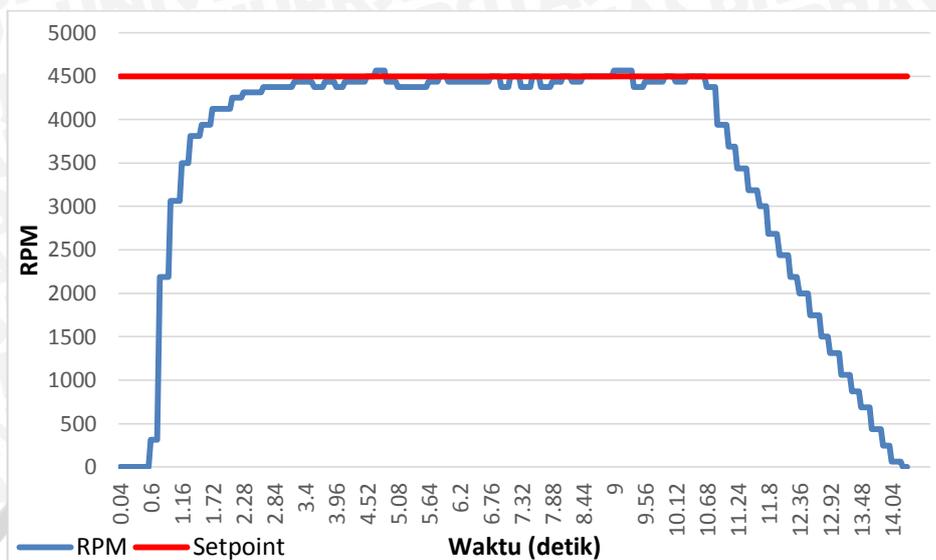
Untuk nilai $K_p = 0.4$, $K_i = 0.1$, dan $K_d = 0$ diperoleh grafik respon seperti pada Gambar 4.12



Gambar 4.12 Grafik Respon untuk $K_p = 0.4$ dan $K_i = 0.1$

$$\begin{aligned} \% E_{ss} &= \frac{|Average\ speed\ steady - setpoint|}{setpoint} \\ &= \frac{|4484.677 - 4500|}{4500} \times 100 \% \\ &= 0.341\% \text{ dan memiliki waktu } steady \text{ 3.08 detik.} \end{aligned}$$

Untuk nilai $K_p = 0.4$, $K_i = 0.3$, dan $K_d = 0$ diperoleh grafik respon seperti pada Gambar 4.13



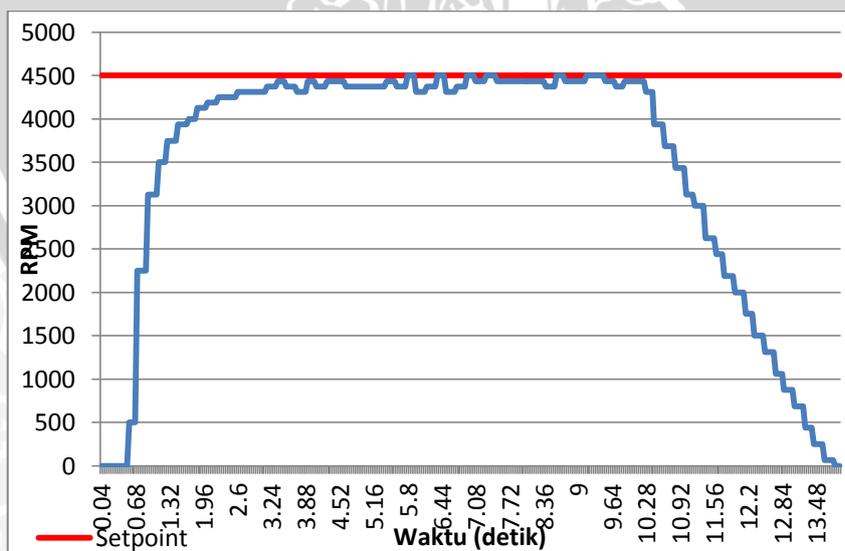
Gambar 4.13 Grafik Respon untuk $K_p = 0.4$ dan $K_i = 0.3$

$$\% E_{ss} = \frac{|Average\ speed\ steady - setpoint|}{setpoint} \times 100\%$$

$$= \frac{|4457.039 - 4500|}{4500} \times 100\%$$

$$= 0.954\% \text{ dan memiliki waktu } steady \text{ 5.68 detik.}$$

Untuk nilai $K_p = 0.4$, $K_i = 0.5$, dan $K_d = 0$ diperoleh grafik respon seperti pada Gambar 4.14



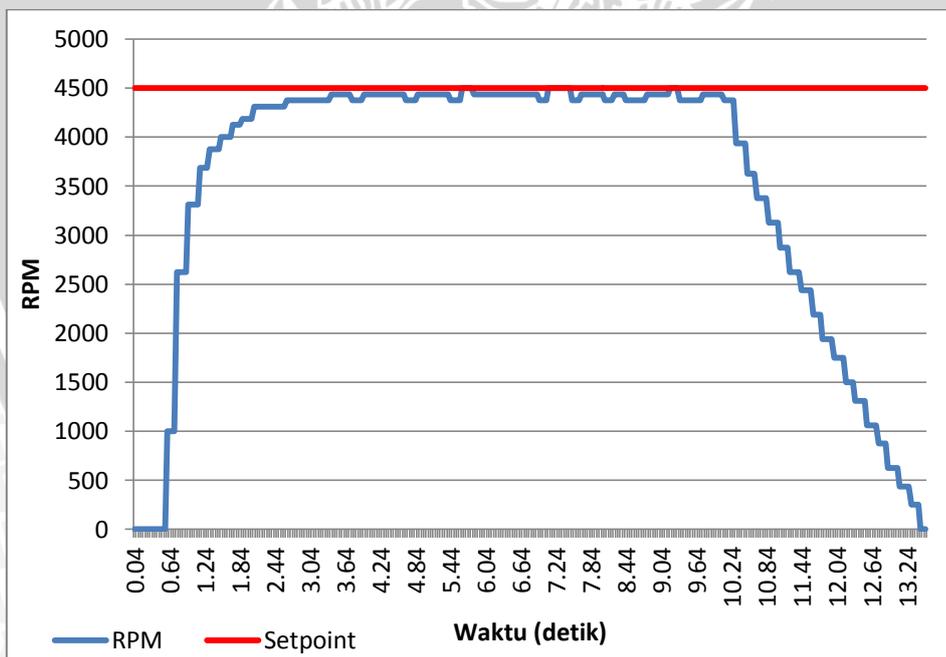
Gambar 4.14 Grafik Respon untuk $K_p = 0.4$ dan $K_i = 0.5$

$$\begin{aligned} \% E_{ss} &= \frac{|Average\ speed\ steady - setpoint|}{setpoint} \times 100 \% \\ &= \frac{|4417.358 - 4500|}{4500} \times 100\% \\ &= 1.836\% \text{ dan memiliki waktu } steady \text{ 6.16 detik.} \end{aligned}$$

Grafik respon yang ditunjukkan mulai dari Gambar 4.12 hingga Gambar 4.14 yang dinyatakan memiliki respon terbaik diantara ketiganya adalah yang ditunjukkan pada Gambar 4.12 dengan nilai parameter $K_i=0.1$ dikarenakan respon tersebut memiliki $\%E_{ss}$ paling kecil.

Selanjutnya untuk mempercepat reaksi sistem, menghilangkan *offset* dan menghasilkan perubahan awal yang besar maka dilakukan *tuning* dengan ketiga parameter PID.

Untuk nilai $K_p = 0.4$, $K_i = 0.1$, dan $K_d = 0.1$ diperoleh grafik respon seperti pada Gambar 4.15



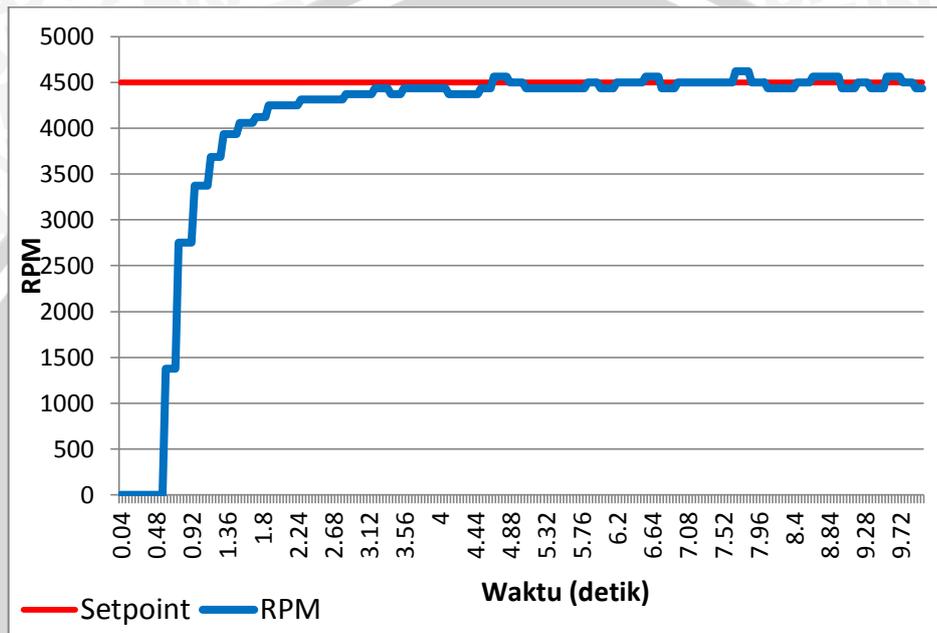
Gambar 4.15 Grafik Respon untuk $K_p = 0.95$, $K_i = 0.1$, dan $K_d = 0.1$

$$\% E_{ss} = \frac{|Average\ speed\ steady - setpoint|}{setpoint} \times 100\%$$

$$= \frac{|4424.535 - 4500|}{4500} \times 100\%$$

$$= 1.633\%$$

Untuk nilai $K_p = 0.4$, $K_i = 0.1$, dan $K_d = 1$ diperoleh grafik respon seperti pada Gambar 4.16



Gambar 4.16 Grafik Respon untuk $K_p = 0.4$, $K_i = 0.1$, dan $K_d = 0.5$

$$\% E_{ss} = \frac{|Average\ speed\ steady - setpoint|}{setpoint} \times 100\%$$

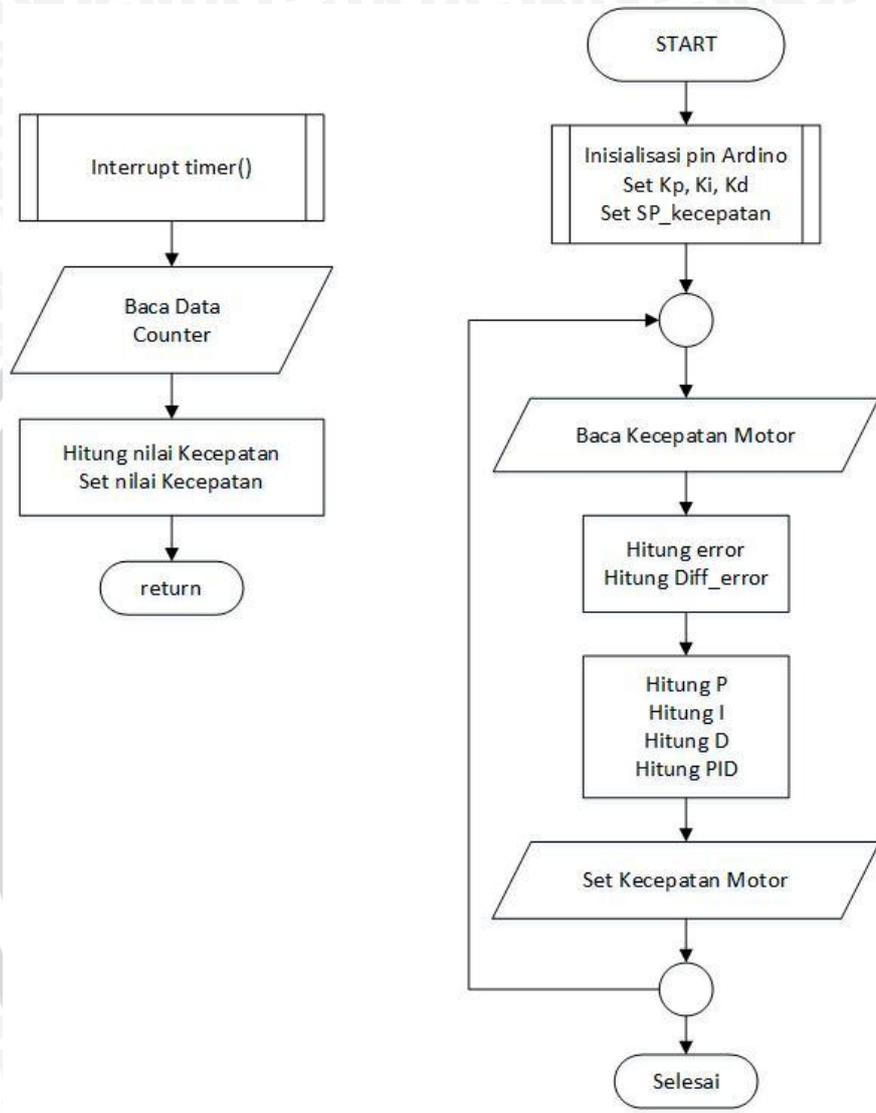
$$= \frac{|4480.3 - 4500|}{4500} \times 100\%$$

$$= 0.4\% \text{ dan memiliki waktu } steady \text{ 3.36 detik.}$$

Hasil *tuning* parameter kontroler telah diperoleh nilai $K_p = 0.4$, $K_i = 0.1$, dan $K_d = 0.5$ didapatkan respon yang baik karena tidak ada *overshoot*, waktu *steady* 3.36 detik dan hanya memiliki *error steady state* 0.4%.

4.4.2. Flowchart

Flowchart perancangan perangkat lunak ditunjukkan pada Gambar 4.17 berikut:



Gambar 4.17 Flowchart Program

Proses dimulai dengan menentukan nilai *setpoint* kecepatan, inisialisasi pin arduino, dan set parameter Kp, Ki, Kd. Kemudian sensor mendeteksi kecepatan motor apabila sudah sesuai dengan nilai *setpoint* proses dinyatakan selesai. Namun apabila respon belum sesuai dengan nilai *setpoint* maka dilakukan perhitungan error dan selisih error serta dilakukan perhitungan parameter P, I, D hingga menghasilkan respon yang sesuai.

BAB V

PENGUJIAN & ANALISIS SISTEM

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

5.1. Pengujian Sensor Kecepatan

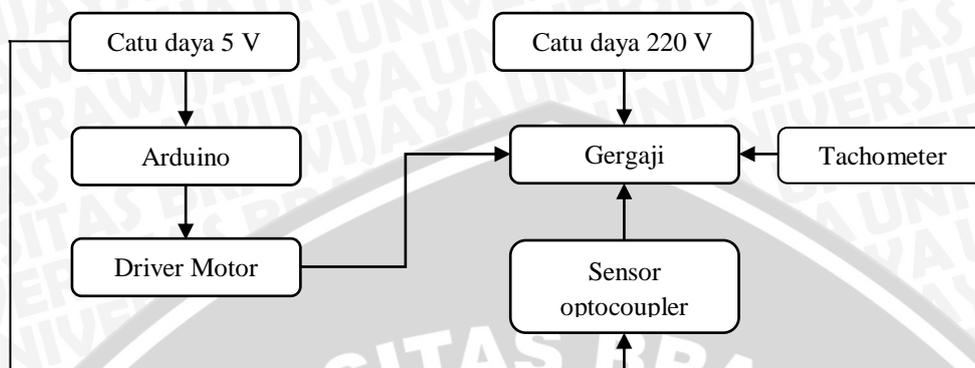
Pengujian sensor kecepatan dilakukan untuk mengetahui RPM yang terbaca pada sensor kecepatan *optocoupler* telah sesuai dengan yang diinginkan.

5.1.1. Peralatan yang digunakan

- Catu daya 220 V
- Sensor kecepatan *optocoupler*
- Gergaji Hitachi Koki C7SS
- Program dan *Software* Arduino
- *Driver* motor

5.1.2. Prosedur Pengujian

1. Merangkai peralatan seperti pada Gambar 5.1
2. Merekam data kecepatan yang diperoleh dari pembacaan *Serial Monitor* program Arduino sebagai *monitor* sensor dan dibandingkan dengan hasil pembacaan dari tachometer.



Gambar 5.1 Skema pengujian sensor Optocoupler

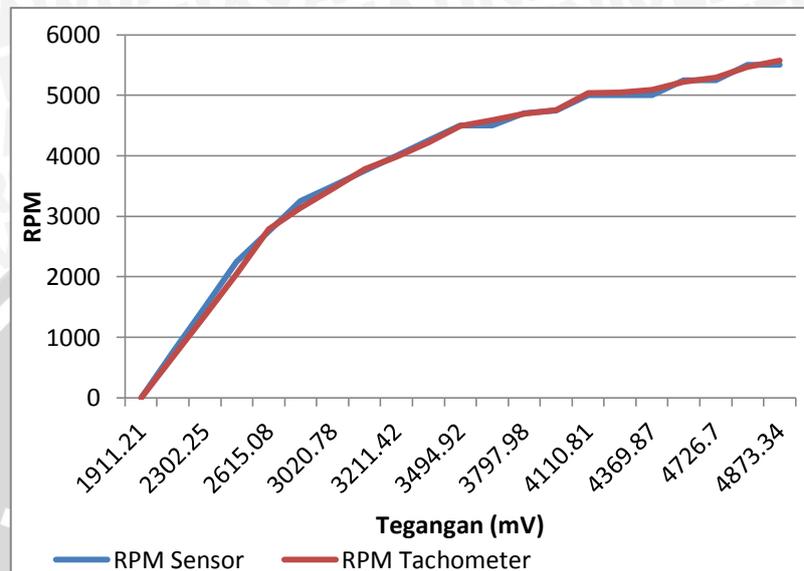
5.1.3. Hasil Pengujian

Setelah melakukan prosedur pengujian sesuai dengan skema, didapatkan data hasil pengujian berupa rpm gergaji dengan nilai tegangan yang ditentukan. Data hasil pengujian ditunjukkan pada Tabel 5.1

Tabel 5.1 Data Hasil Pengujian Sensor Optocoupler

RPM Sensor	RPM Tachometer
0	0
750	678
1500	1358
2250	2048
2750	2789
3250	3139
3500	3450
3750	3780
4000	3980
4000	4220
4500	4495
4500	4589
4750	4698
4750	4755
5000	5036
5000	5048
5000	5089
5250	5220
5250	5289
5500	5467
5500	5570

Dari Tabel 5.1 diperoleh grafik hubungan antara tegangan potensio dan rpm gergaji seperti pada Gambar 5.2.



Gambar 5.2 Grafik Perbandingan Nilai Pembacaan Sensor Kecepatan dengan Data Praktek

Dari Gambar 5.2 didapatkan data perbandingan antara RPM dan tegangan. Data tegangan yang dimaksud adalah tegangan input untuk driver motor yang terhubung pada gergaji. Sedangkan RPM adalah kecepatan putar motor gergaji dalam satuan rpm.

Dari hasil pengujian yang dilakukan dapat disimpulkan bahwa hasil pembacaan sensor dapat dikatakan mendekati hasil pembacaan tachometer sebagai acuan pengujian data kecepatan. Selain itu terbukti pula dari hasil pengujian apabila tegangan naik maka kecepatan gergaji akan naik pula.

5.2. Pengujian Motor Gergaji

Pengujian kecepatan gergaji dilakukan untuk mengetahui hubungan tegangan input *driver* motor dengan kecepatan motor.

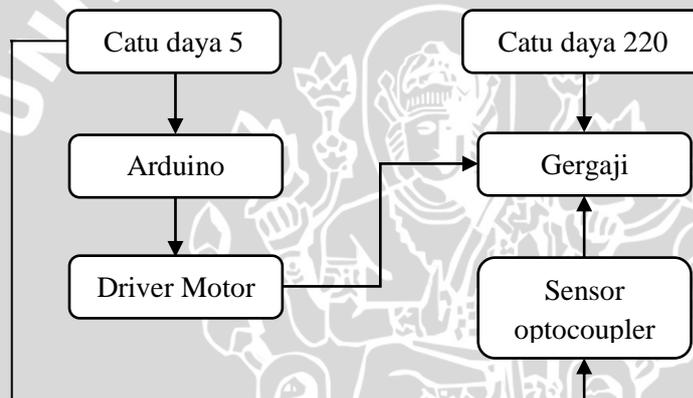
5.2.1. Peralatan yang digunakan

- Catu daya 220 V

- Sensor kecepatan *optocoupler*
- Gergaji Hitachi Koki C7SS
- Program dan *Software* Arduino
- *Driver* motor

5.2.2. Prosedur Pengujian

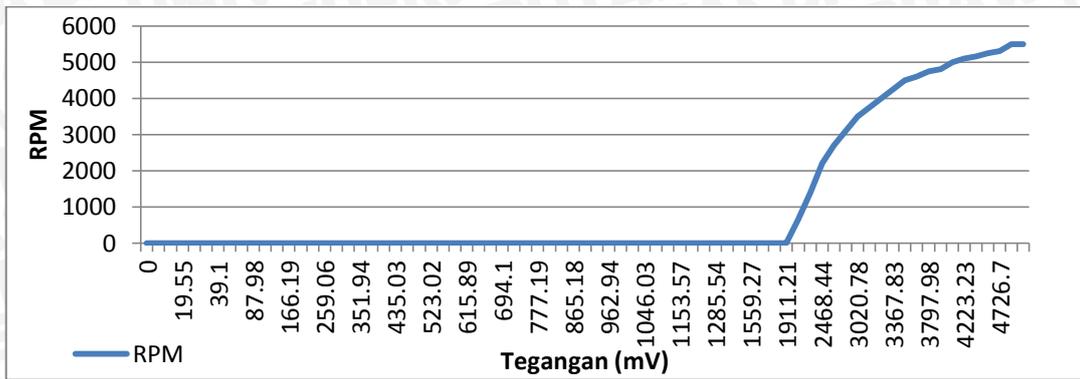
1. Merangkai peralatan seperti pada Gambar 5.3
2. Mengaktifkan catu daya 220 V
3. Mencatat data kecepatan yang diperoleh dari pembacaan *serial monitor* Arduino.



Gambar 5.3 Rangkaian Pengujian Kecepatan Motor

5.2.3. Hasil Pengujian

Setelah melakukan prosedur pengujian, didapatkan data hasil pengujian berupa respon kecepatan motor gergaji dibandingkan dengan tegangan input *driver* motor yang ditunjukkan pada Gambar 5.4



Gambar 5.4 Grafik Hubungan Tengan dengan Kecepatan Motor

Dapat dilihat pada Gambar 5.4 motor gergaji mulai berputar di kisaran tegangan 2 volt, kecepatan motor gergaji mengalami perubahan yang besar dalam kisaran tegangan 2V-3V sedangkan pada kisaran tegangan 3V-5V perubahan kecepatan tidak begitu besar.

5.3. Pengujian Blok Kontroler

5.3.1. Pengujian tanpa Kontroler

Pengujian ini bertujuan untuk mengetahui bagaimana perbedaan respon sistem terhadap *setpoint* ketika menggunakan kontroler dan tanpa kontroler, sehingga dapat ditentukan bahwa sistem memerlukan sebuah sistem pengendalian. Pengujian ini masih dengan peralatan dan prosedur yang sama dengan pengujian motor gergaji, akan tetapi nilai parameter Kp, Ki, dan Kd tidak dimasukkan. Berikut Tabel 5.2 data pengujian tanpa Kontroler.

Tabel 5.2 Data Pengujian tanpa Kontroler

Setpoint	Kecepatan Motor (rpm)
4500	0
4500	0
4500	0
4500	0
4500	0
4500	0
4500	0
4500	0
4500	0
4500	0

<i>Set Point</i>	Kecepatan Motor (rpm)
4500	0
4500	0
4500	0
4500	0
4500	0
4500	0
4500	0
4500	187
4500	187
4500	187
4500	187
4500	187
4500	1500
4500	1500
4500	1500
4500	1500
4500	2375
4500	2375
4500	2375
4500	2937
4500	2937
4500	2937
4500	2937
4500	3312
4500	3312
4500	3312
4500	3437
4500	3437
4500	3437
4500	3437
4500	3437
4500	3625
4500	3625
4500	3625
4500	3625
4500	3750
4500	3750
4500	3750

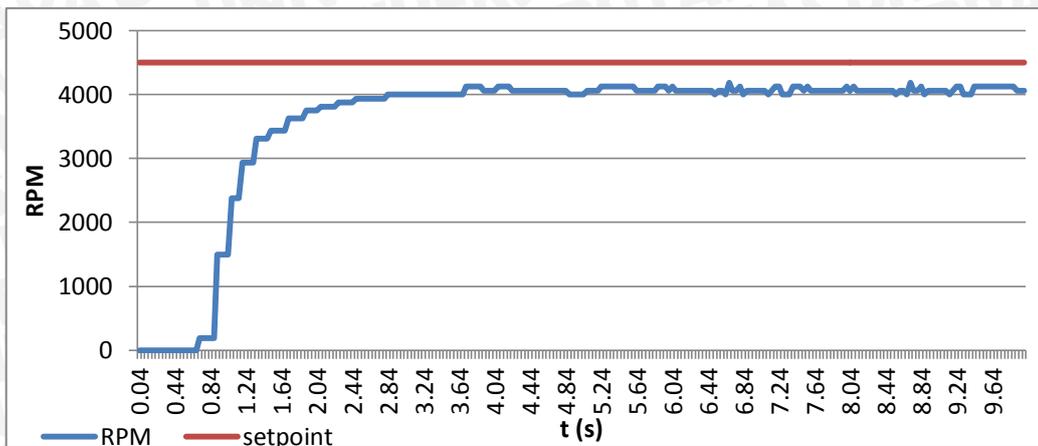


Set Point	Kecepatan Motor (rpm)
4500	4000
4500	4000
4500	4125
4500	4125
4500	4125
4500	4125
4500	4125
4500	4062
4500	4062
4500	4062
4500	4062
4500	4125
4500	4125
4500	4125
4500	4125
4500	4062
4500	4062
4500	4062
4500	4062
4500	4062
4500	4062
4500	4062
4500	4062
4500	4062
4500	4062
4500	4062
4500	4062
4500	4062
4500	4062
4500	4062
4500	4000
4500	4000
4500	4000
4500	4000
4500	4062
4500	4062
4500	4062



<i>Set Point</i>	Kecepatan Motor (rpm)
4500	4125
4500	4125
4500	4125
4500	4125
4500	4125
4500	4125
4500	4125
4500	4125
4500	4125
4500	4125
4500	4062
4500	4062
4500	4062
4500	4062
4500	4062
4500	4062
4500	4125
4500	4125
4500	4125
4500	4062
4500	4125
4500	4062
4500	4062
4500	4062
4500	4062
4500	4062
4500	4062
4500	4062
4500	4062
4500	4062
4500	4062
4500	4062
4500	4062
4500	4000
4500	4062
4500	4062
4500	4187
4500	4062
4500	4062
4500	4125





Gambar 5.5 Grafik Pengujian tanpa Kontroler

Respon ketika tanpa kontroler menunjukkan bahwa kecepatan motor tidak dapat mencapai nilai *setpoint* yang ditentukan yaitu 4500 rpm, seperti yang ditunjukkan pada gambar 5.5. Maka agar dapat mencapai *setpoint* perlu ditambahkan kontroler untuk mengendalikan kecepatan motor.

5.3.2. Pengujian dengan Kontroler

Pengujian ini bertujuan untuk mengetahui respon sistem ketika diberi kontroler. Dengan alat dan langkah prosedur pengujian yang sama seperti pada pengujian motor dilakukan pengambilan data respon kecepatan motor ketika menggunakan kontroler. Pada tabel 5.3 terlihat data pengujian sistem menggunakan kontroler.

Tabel 5.3 Data Pengujian Menggunakan Kontroler

<i>Setpoint</i>	<i>Kecepatan Motor (rpm)</i>
4500	0
4500	0
4500	0
4500	0
4500	0
4500	0
4500	0
4500	0
4500	0
4500	0

<i>Set Point</i>	Kecepatan Motor (rpm)
4500	0
4500	0
4500	0
4500	0
4500	1375
4500	1375
4500	1375
4500	1375
4500	2750
4500	2750
4500	2750
4500	2750
4500	2750
4500	3375
4500	3375
4500	3375
4500	3375
4500	3375
4500	3687
4500	3687
4500	3687
4500	3687
4500	3937
4500	3937
4500	3937
4500	3937
4500	3937
4500	4062
4500	4062
4500	4062
4500	4062
4500	4062
4500	4125
4500	4125
4500	4125
4500	4125
4500	4250
4500	4250
4500	4250
4500	4250

<i>Set Point</i>	Kecepatan Motor (rpm)
4500	4250
4500	4250
4500	4250
4500	4250
4500	4250
4500	4250
4500	4312
4500	4312
4500	4312
4500	4312
4500	4312
4500	4312
4500	4312
4500	4312
4500	4312
4500	4312
4500	4312
4500	4312
4500	4312
4500	4312
4500	4312
4500	4312
4500	4375
4500	4375
4500	4375
4500	4375
4500	4375
4500	4375
4500	4375
4500	4375
4500	4375
4500	4375
4500	4437
4500	4437
4500	4437
4500	4437
4500	4437
4500	4375
4500	4375
4500	4375
4500	4437
4500	4437



<i>Set Point</i>	Kecepatan Motor (rpm)
4500	4437
4500	4437
4500	4437
4500	4437
4500	4437
4500	4437
4500	4437
4500	4437
4500	4437
4500	4437
4500	4437
4500	4375
4500	4375
4500	4375
4500	4375
4500	4375
4500	4375
4500	4375
4500	4375
4500	4375
4500	4375
4500	4437
4500	4437
4500	4437
4500	4562
4500	4562
4500	4562
4500	4562
4500	4562
4500	4500
4500	4500
4500	4500
4500	4437
4500	4437
4500	4437

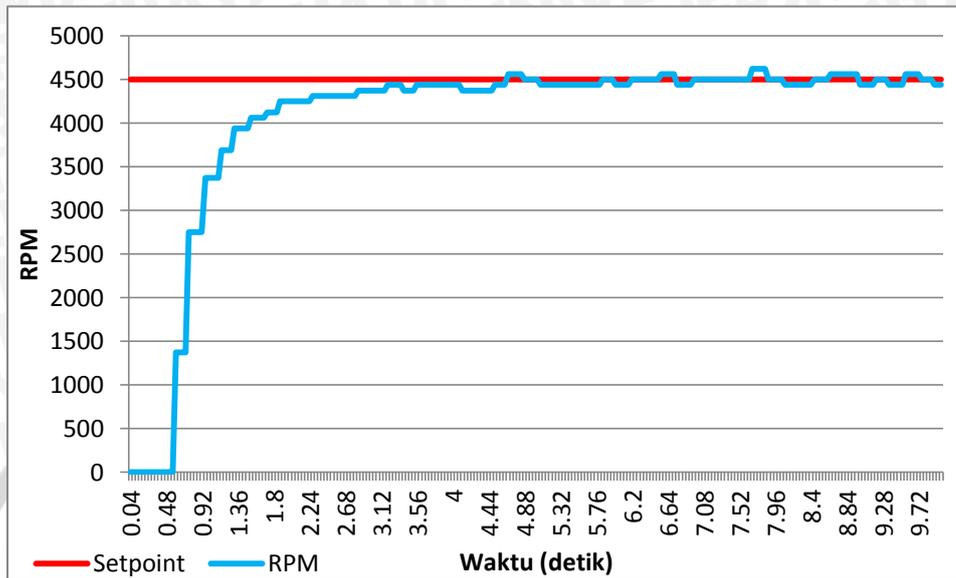


<i>Set Point</i>	Kecepatan Motor (rpm)
4500	4437
4500	4437
4500	4437
4500	4437
4500	4437
4500	4437
4500	4437
4500	4437
4500	4437
4500	4437
4500	4437
4500	4437
4500	4437
4500	4437
4500	4437
4500	4500
4500	4500
4500	4500
4500	4500
4500	4437
4500	4437
4500	4437
4500	4437
4500	4437
4500	4500
4500	4500
4500	4500
4500	4500
4500	4500
4500	4500
4500	4500
4500	4500
4500	4500
4500	4500
4500	4562
4500	4562
4500	4562
4500	4562
4500	4562
4500	4437
4500	4437



<i>Set Point</i>	Kecepatan Motor (rpm)
4500	4500
4500	4500
4500	4500
4500	4500
4500	4500
4500	4562
4500	4562
4500	4562
4500	4562
4500	4562
4500	4562
4500	4562
4500	4562
4500	4437
4500	4437
4500	4437
4500	4437
4500	4437
4500	4500
4500	4500
4500	4500
4500	4500
4500	4437
4500	4437
4500	4437
4500	4437
4500	4437
4500	4562
4500	4562
4500	4562
4500	4562
4500	4500
4500	4500
4500	4500
4500	4437
4500	4437
4500	4437





Gambar 5.6 Grafik Pengujian dengan Kontroler

Pada gambar 5.6 terlihat bahwa apabila sistem menggunakan kontroler maka kecepatan dapat mencapai nilai *setpoint* yang ditentukan yaitu 4500 rpm. Data pada gambar 5.6 didapat dari pengujian secara *hand tuning* setelah beberapa kali pengujian. Dengan nilai parameter $K_p=0.4$, $K_i=0.1$, dan $K_d=0.5$ sehingga respon yang dihasilkan dapat mencapai keadaan yang diinginkan yaitu mencapai *setpoint* 4500 rpm.

5.4. Pengujian Keseluruhan Sistem

Pengujian ini bertujuan untuk mengetahui bagaimana kinerja sistem secara keseluruhan dan mengamati respon kontroler terhadap *setpoint* ketika mendapat beban atau gangguan berupa kayu. Untuk pengujian dibutuhkan kayu dengan tingkat kekerasan yang berbeda. Agar mempermudah pengujian kayu yang digunakan harus memiliki ukuran lebar maksimal 10 cm dan tinggi maksimal 7 cm atau dapat menggunakan potongan kayu yang biasa digunakan untuk kerangka atap rumah. Pengujian berbeban menggunakan 3 jenis kayu dengan tingkat kekerasan yang berbeda, yaitu kayu Albasia sebagai beban 1, kayu Mahoni sebagai beban 2, kayu Jati sebagai beban 3.

Berikut tabel 5.4 menunjukkan data hasil percobaan dengan pengujian beban 3 jenis kayu.

Tabel 5.4 Data Pengujian Keseluruhan dengan Beban 3 Jenis Kayu

<i>Setpoint</i>	Beban Kayu Albasia	Beban Kayu Mahoni	Beban Kayu Jati
4500	0	0	0
4500	0	0	0
4500	0	0	0
4500	0	0	0
4500	0	0	0
4500	0	0	0
4500	0	0	0
4500	625	62	62
4500	625	62	62
4500	625	62	62
4500	625	62	62
4500	625	1750	1750
4500	2375	1750	1750
4500	2375	1750	1750
4500	2375	1750	1750
4500	3125	2937	2812
4500	3125	2937	2812
4500	3125	2937	2812
4500	3125	2937	2812
4500	3562	3437	3375
4500	3562	3437	3375
4500	3562	3437	3375
4500	3562	3437	3375
4500	3562	3750	3687
4500	3750	3750	3687
4500	3750	3750	3687
4500	3750	3750	3687
4500	3750	3750	3687
4500	3750	3937	3875
4500	4000	3937	3875
4500	4000	3937	3875
4500	4000	3937	3875
4500	4062	4125	4062
4500	4062	4125	4062
4500	4062	4125	4062
4500	4062	4125	4062
4500	4062	4187	4187
4500	4187	4187	4187
4500	4187	4187	4187
4500	4187	4187	4187
4500	4187	4187	4187
4500	4187	4312	4250
4500	4250	4312	4250
4500	4250	4312	4250

<i>Setpoint</i>	Beban Kayu Albasia	Beban Kayu Mahoni	Beban Kayu Jati
4500	4437	4437	4437
4500	4437	4437	4437
4500	4437	4437	4437
4500	4437	4500	4500
4500	4437	4500	4500
4500	4437	4500	4500
4500	4437	4500	4500
4500	4437	4500	4500
4500	4500	4500	4500
4500	4500	4500	4500
4500	4500	4500	4437
4500	4500	4500	4437
4500	4500	4500	4437
4500	4500	4500	4437
4500	4500	4500	4500
4500	4500	4437	4500
4500	4375	4437	4500
4500	4375	4437	4500
4500	4375	4437	4500
4500	4375	4437	4500
4500	4375	4500	4500
4500	4437	4500	4500
4500	4437	4500	4500
4500	4437	4500	4500
4500	4437	4500	4500
4500	4437	4500	4500
4500	4437	4500	4500
4500	4437	4500	4500
4500	4437	4500	4500
4500	4437	4500	4500
4500	4437	4437	4375
4500	4375	4500	4375
4500	4375	4500	4375
4500	4375	4500	4375
4500	4375	4500	4375
4500	4375	4500	4375
4500	4375	4500	4437
4500	4437	4500	4437
4500	4437	4475	4437
4500	4437	4475	4437
4500	4437	4475	4437
4500	4375	4475	4437
4500	4375	4500	4437
4500	4375	4500	4437
4500	4375	4500	4375
4500	4250	4500	4375
4500	4250	4500	4375
4500	4250	4500	4375
4500	4250	4500	4375
4500	4250	4500	4375
4500	4250	4500	4437
4500	4187	4500	4437
4500	4187	4437	4437



<i>Setpoint</i>	Beban Kayu Albasia	Beban Kayu Mahoni	Beban Kayu Jati
4500	4187	4437	4437
4500	4187	4437	4375
4500	4000	4437	4375
4500	4000	4500	4500
4500	4000	4500	4500
4500	4000	4500	4500
4500	4000	4500	4500
4500	3812	4500	4500
4500	3812	4500	4500
4500	3812	4500	4437
4500	3812	4500	4437
4500	3812	4500	4437
4500	3687	4500	4437
4500	3687	4500	4437
4500	3687	4500	4312
4500	3687	4500	4312
4500	3625	4500	4312
4500	3625	4437	4312
4500	3625	4437	4312
4500	3625	4437	4125
4500	3625	4437	4125
4500	3562	4500	4125
4500	3562	4500	4125
4500	3562	4500	4000
4500	3562	4500	4000
4500	3562	4500	4000
4500	3750	4500	4000
4500	3750	4375	4000
4500	3750	4375	3812
4500	3750	4375	3812
4500	3750	4375	3812
4500	4000	4375	3812
4500	4000	4437	3812
4500	4000	4437	3812
4500	4000	4437	3812
4500	4062	4437	3812
4500	4062	4437	3812
4500	4062	4437	3687
4500	4062	4437	3687
4500	4062	4437	3687
4500	4187	4437	3687
4500	4187	4375	3687
4500	4187	4375	3687
4500	4312	4375	3400
4500	4312	4375	3400
4500	4312	4437	3400
4500	4312	4437	3350
4500	4600	4437	3350
4500	4600	4437	3350
4500	4600	4375	3350
4500	4600	4375	3000
4500	4375	4375	3000

<i>Setpoint</i>	Beban Kayu Albasia	Beban Kayu Mahoni	Beban Kayu Jati
4500	4375	4375	3000
4500	4375	4375	3000
4500	4375	4250	3000
4500	4375	4250	3375
4500	4375	4250	3375
4500	4375	4250	3375
4500	4375	4250	3375
4500	4375	4187	3687
4500	4375	4187	3687
4500	4437	4187	3687
4500	4437	4187	3687
4500	4437	4000	3687
4500	4437	4000	3875
4500	4375	4000	3875
4500	4375	4000	3875
4500	4375	4000	3875
4500	4375	3812	3875
4500	4375	3812	4062
4500	4375	3812	4062
4500	4375	3812	4062
4500	4375	3812	4062
4500	4375	3687	4187
4500	4375	3687	4187
4500	4375	3687	4187
4500	4375	3687	4187
4500	4375	3625	4187
4500	4375	3625	4250
4500	4375	3625	4312
4500	4437	3625	4312
4500	4437	3625	4312
4500	4437	3562	4312
4500	4437	3562	4312
4500	4375	3562	4437
4500	4375	3562	4437
4500	4375	3562	4437
4500	4375	3500	4437
4500	4375	3500	4437
4500	4437	3268	4610
4500	4437	3268	4610
4500	4437	3437	4610
4500	4437	3437	4610
4500	4437	3437	4610
4500	4437	3437	4610
4500	4437	3750	4375
4500	4437	3750	4375
4500	4437	3750	4375
4500	4437	3750	4375
4500	4437	3750	4375
4500	4437	3937	4375
4500	4437	3937	4375
4500	4437	3937	4375
4500	4437	3937	4437
4500	4437	3937	4437



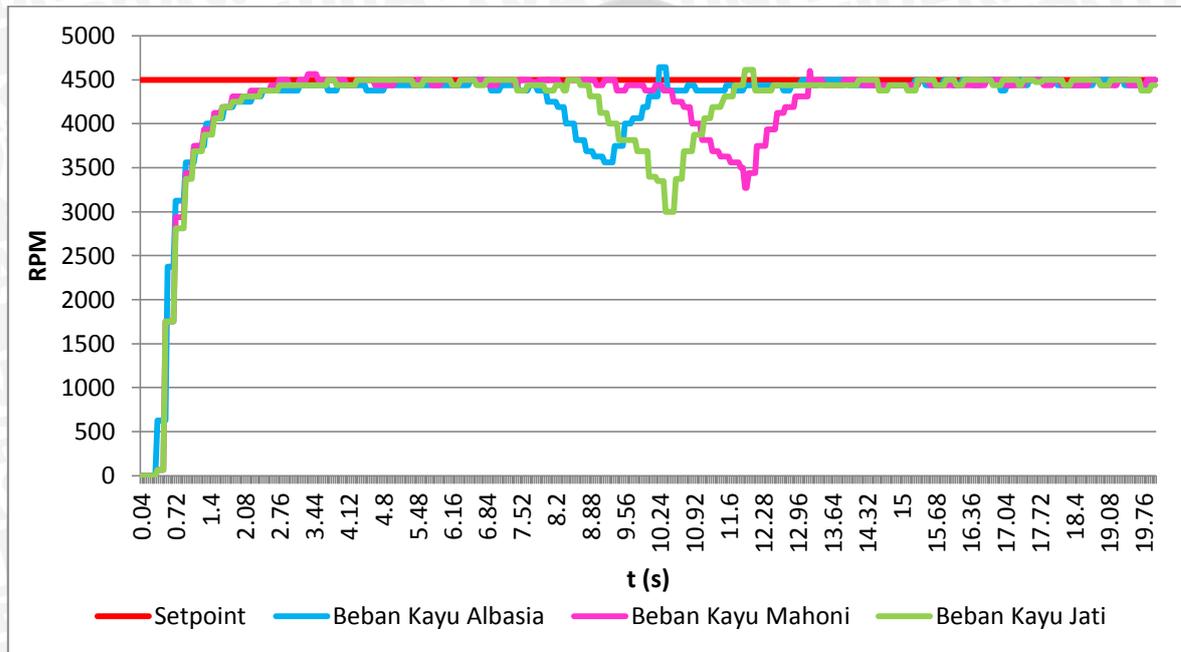
<i>Setpoint</i>	Beban Kayu Albasia	Beban Kayu Mahoni	Beban Kayu Jati
4500	4437	4125	4437
4500	4437	4125	4437
4500	4437	4125	4437
4500	4375	4125	4437
4500	4375	4187	4437
4500	4375	4187	4437
4500	4375	4187	4437
4500	4437	4187	4437
4500	4437	4187	4437
4500	4437	4312	4437
4500	4437	4312	4437
4500	4437	4312	4437
4500	4500	4312	4437
4500	4500	4312	4437
4500	4500	4312	4437
4500	4500	4600	4437
4500	4437	4437	4437
4500	4437	4437	4437
4500	4437	4500	4437
4500	4437	4500	4437
4500	4500	4500	4437
4500	4500	4500	4437
4500	4500	4437	4437
4500	4500	4437	4437
4500	4500	4437	4437
4500	4500	4437	4437
4500	4500	4437	4437
4500	4500	4437	4437
4500	4500	4437	4437
4500	4500	4437	4437
4500	4500	4437	4437
4500	4500	4437	4437
4500	4500	4500	4437
4500	4500	4500	4437
4500	4437	4500	4437
4500	4437	4500	4437
4500	4437	4500	4437
4500	4437	4500	4437
4500	4437	4500	4437
4500	4437	4500	4437
4500	4437	4500	4437
4500	4437	4500	4437
4500	4437	4500	4437
4500	4437	4437	4500
4500	4437	4437	4500
4500	4437	4437	4500
4500	4437	4437	4500
4500	4437	4437	4375
4500	4437	4437	4375
4500	4437	4437	4375



<i>Setpoint</i>	Beban Kayu Albasia	Beban Kayu Mahoni	Beban Kayu Jati
4500	4500	4500	4437
4500	4500	4500	4437
4500	4500	4500	4437
4500	4500	4500	4437
4500	4375	4500	4437
4500	4375	4500	4500
4500	4375	4437	4500
4500	4437	4437	4500
4500	4437	4437	4500
4500	4437	4437	4500
4500	4500	4437	4500
4500	4500	4437	4500
4500	4500	4437	4437
4500	4500	4437	4437
4500	4437	4500	4500
4500	4437	4500	4500
4500	4500	4500	4500
4500	4500	4500	4500
4500	4500	4500	4500
4500	4500	4500	4500
4500	4500	4500	4475
4500	4500	4500	4475
4500	4500	4500	4475
4500	4500	4500	4475
4500	4500	4437	4500
4500	4500	4437	4500
4500	4500	4437	4500
4500	4500	4437	4500
4500	4500	4500	4500
4500	4500	4500	4500
4500	4437	4500	4500
4500	4437	4500	4500
4500	4437	4437	4500
4500	4437	4437	4437
4500	4437	4437	4437
4500	4437	4437	4437
4500	4437	4437	4437
4500	4437	4437	4500
4500	4437	4437	4500
4500	4437	4437	4500
4500	4437	4437	4500
4500	4437	4437	4500
4500	4437	4437	4500
4500	4437	4437	4500
4500	4437	4437	4500
4500	4500	4500	4500
4500	4500	4500	4500
4500	4500	4500	4500
4500	4500	4500	4500



Berikut gambar 5.7 menunjukkan data hasil pengujian dengan beban 3 jenis kayu dalam bentuk grafik.



Gambar 5.7 Data Grafik Pengujian Berbeban

Dengan nilai parameter $K_p=0.4$, $K_i=0.1$, dan $K_d=0.5$ dilakukan pengujian dengan 3 jenis beban berupa Kayu Albasia yang mewakili kayu dengan tingkat kekerasan lunak, Kayu Mahoni yang mewakili kayu dengan tingkat kekerasan sedang, dan Kayu Jati yang mewakili kayu dengan tingkat kekerasan paling tinggi diantara ketiganya. Respon sistem ketika diberi gangguan dapat dilihat pada gambar 5.7

Dari pengujian berbeban seperti yang terlihat pada gambar 5.7 didapat $\%E_{ss}$ (persentase *error steady state*) dan $\%M_p$ (persentase maksimum *overshoot*) dengan perhitungan sebagai berikut:

- Untuk beban berupa Kayu Albasia

$$\%E_{ss} = \frac{|Average\ speed\ steady - setpoint|}{setpoint} \times 100\%$$

$$= \frac{|4458.14 - 4500|}{4500} \times 100\%$$

$$= \frac{41.86}{4500} \times 100\%$$

$$= 0.00930222 \times 100\%$$

$$= 0.930222\%$$

$$\%M_p = \frac{\text{max speed} - \text{setpoint}}{\text{setpoint}} \times 100\%$$

$$= \frac{4645 - 4500}{4500} \times 100\%$$

$$= \frac{145}{4500} \times 100\%$$

$$= 0.03222 \times 100\%$$

$$= 3.222\%$$

- Untuk beban berupa Kayu Mahoni

$$\%E_{ss} = \frac{|Average\ speed\ steady - setpoint|}{setpoint} \times 100\%$$

$$= \frac{|4452.466 - 4500|}{4500} \times 100\%$$

$$= \frac{47.534}{4500} \times 100\%$$

$$= 0.01056311 \times 100\%$$

$$= 1.056311\%$$

$$\%M_p = \frac{\text{max speed} - \text{set point}}{\text{set point}} \times 100\%$$

$$= \frac{4600 - 4500}{4500} \times 100\%$$

$$= \frac{100}{4500} \times 100\%$$

$$= 0.0222 \times 10\%$$

$$= 2.222\%$$

- Untuk beban berupa Kayu Jati

$$\%E_{ss} = \frac{|Average\ speed\ steady - setpoint|}{setpoint} \times 100\%$$
$$= \frac{|4469.175 - 4500|}{4500} \times 100\%$$

$$= \frac{30.825}{4500} \times 100\%$$

$$= 0.00685 \times 100\%$$

$$= 0.685\%$$

$$\%M_p = \frac{max\ speed - setpoint}{setpoint} \times 100\%$$

$$= \frac{4610}{4500} \times 100\%$$

$$= \frac{110}{4500} \times 100\%$$

$$= 0.02444 \times 100\%$$

$$= 2.444\%$$



BAB VI

PENUTUP

6.1. Kesimpulan

Dari perancangan, pengujian dan pengamatan yang telah dilakukan pada penelitian sistem pengendalian kecepatan motor pemotong kayu maka dapat diambil kesimpulan sebagai berikut :

- 1) Berdasarkan data respon sistem yang diperoleh dari pengujian dengan menggunakan metode Ziegler-Nichols ke-2 respon menunjukkan ketidakstabilan atau dengan kata lain mengalami osilasi, maka dilakukan pengujian parameter kontroler PID dengan menggunakan metode *hand tuning* dan mendapatkan nilai $K_p = 0.4$, $K_i = 0.1$, dan $K_d = 0.5$ dan toleransi error sebesar 0.4%.
- 2) Hasil pengujian terhadap sistem pengendalian kecepatan putaran motor pemotong kayu menunjukkan bahwa respon sistem untuk mencapai t_s (*time steady*) membutuhkan waktu 3.36 detik. Jadi hasil pengujian ini menunjukkan bahwa Arduino uno dengan metode kontrol PID menghasilkan respon sesuai dengan yang diharapkan dan mampu diaplikasikan pada pemotong kayu dengan baik, terbukti hasil pengujian berbeban menunjukkan %Ess = 0.930222% dan %Mp = 3.222% untuk beban Kayu Albasia, %Ess = 1.056311% dan %Mp = 2.222% untuk beban kayu Mahoni, dan %Ess = 0.685% dan %Mp = 2.444% untuk beban Kayu Jati

DAFTAR PUSTAKA

Amstrom, K. J., & Hagglund, Tore. 1995 *PID Controllers: Theory, Design and Tuning*. Instrument Society of America: Research Triangle Park.

Artanto, Dian. 2012. *Interaksi Arduino dan LabVIEW*. Jakarta: PT Gramedia.

Barley, Eres. 2011. *Sistem Pengaturan Kecepatan Putaran Motor DC Pada Konveyor Bertingkat Menggunakan Kontroler PID Berbasis*

Ogata, Katsuhiko. 1997. *Teknik Kontrol Automatik (Sistem Pengaturan)*. Jakarta: Erlangga.

<http://electronical-instrument.blogspot.com/2010/06/komunikasi-serial-rs232.html> Diakses tanggal 21 Desember 2015

<http://www.hitachi.com/> Diakses tanggal 21 Desember 2015



UNIVERSITAS BRAWIJAYA

LAMPIRAN I

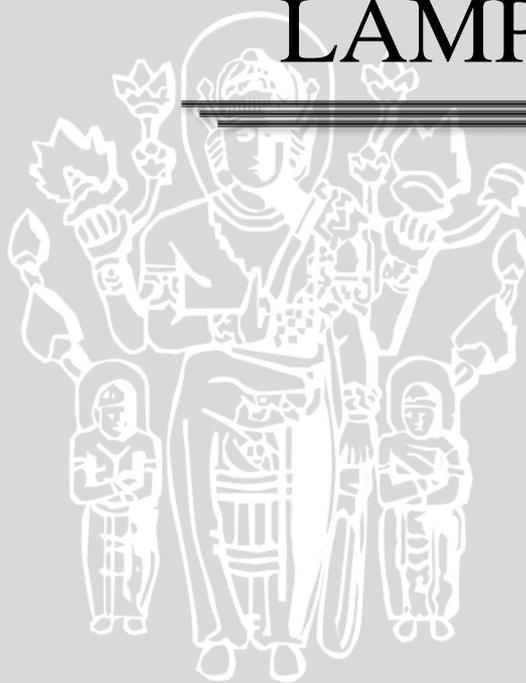


Foto Alat

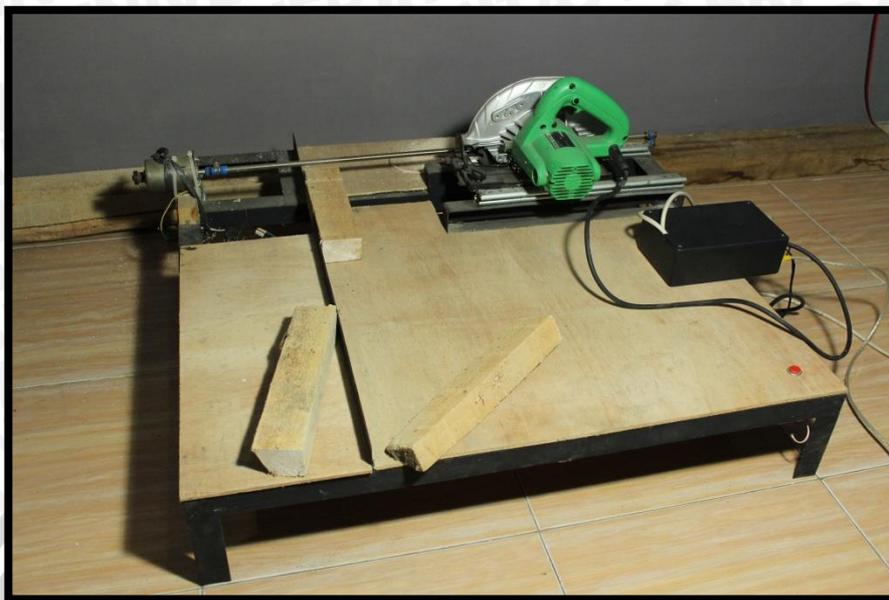
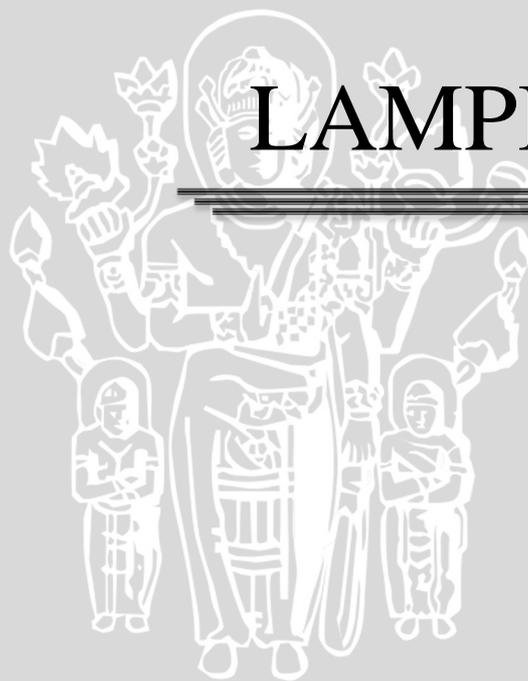


Foto Mekanik Pemotong Kayu



Foto Proses Pembuatan Pemotong Kayu

UNIVERSITAS BRAWIJAYA

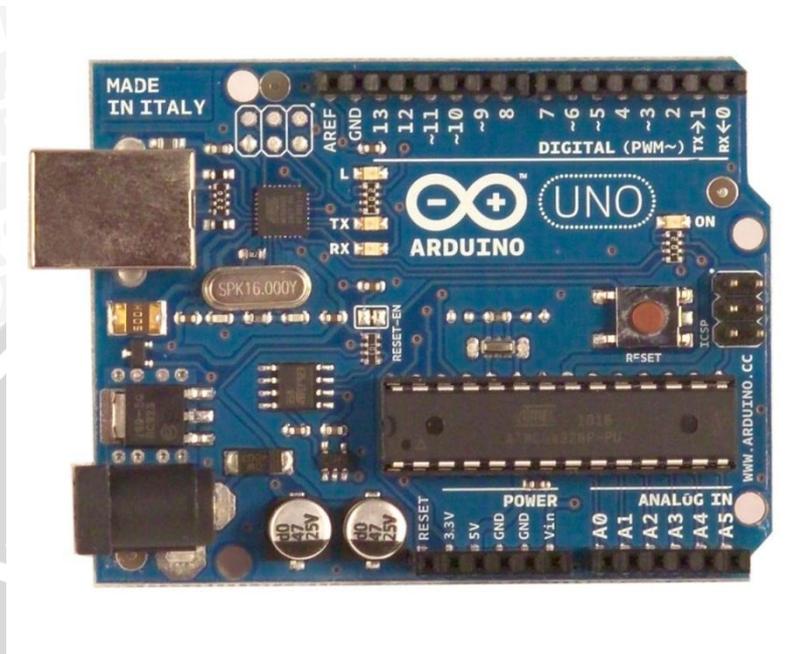


LAMPIRAN II

Data Sheet



Arduino UNO



Product Overview

The Arduino Uno is a microcontroller board based on the ATmega328 ([datasheet](#)). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

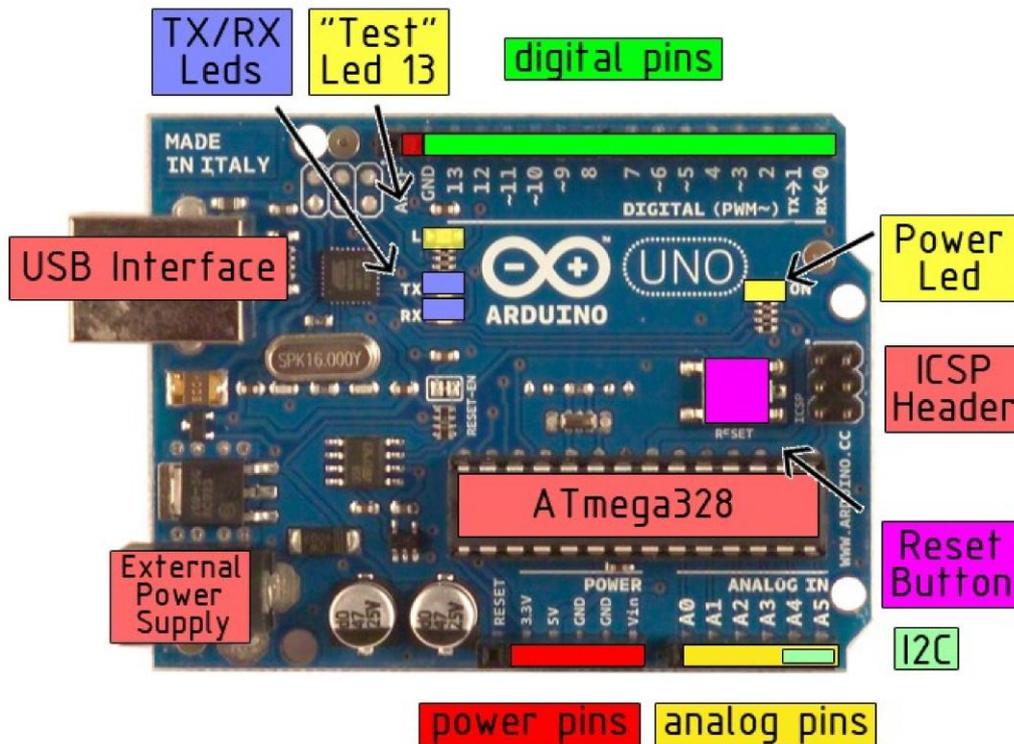
"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version

1.1 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the [index of Arduino boards](#).

Summary

Microcontroller	ATmega328
Operating Voltage	5V Input Voltage (recommended) 7-12V Input Voltage (limits) 6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA DC Current for 3.3V Pin 50 mA
Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

the board



radiospares

RADIONICS



Power

The Arduino Uno can be powered via the USB connection or with an external power supply.

The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery.

The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The Atmega328 has 32 KB of flash memory for storing code (of which 0,5 KB is used for the bootloader); It has also 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip .
- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the [analogReference\(\)](#) function. Additionally, some pins have specialized functionality:

- **I²C: 4 (SDA) and 5 (SCL).** Support I²C (TWI) communication using the [Wire library](#).

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with [analogReference\(\)](#).
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.



radiospares

RADIONICS



Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega8U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '8U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, an *.inf file is required..

The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Uno's digital pins.

The ATmega328 also support I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the [documentation](#) for details. To use the SPI communication, please see the ATmega328 datasheet.

Programming

The Arduino Uno can be programmed with the Arduino software ([download](#)). Select "Arduino Uno w/ ATmega328" from the **Tools > Board** menu (according to the microcontroller on your board). For details, see the [reference](#) and [tutorials](#).

The ATmega328 on the Arduino Uno comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.

The ATmega8U2 firmware source code is available . The ATmega8U2 is loaded with a DFU bootloader, which can be activated by connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2. You can then use [Atmel's FLIP software](#)

(Windows) or the [DFU programmer](#) (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader).



radiospares

RADIONICS



Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

USB Overcurrent Protection

The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics



radiospares

RADIONICS



The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

How to use Arduino



Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the [Arduino programming language](#) (based on [Wiring](#)) and the Arduino development environment (based on [Processing](#)). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

Arduino is a cross-platform program. You'll have to follow different instructions for your personal OS. Check on the [Arduino site](#) for the latest instructions.

<http://arduino.cc/en/Guide/HomePage>

Linux Install

Windows Install

Mac Install

Once you have downloaded/unzipped the arduino IDE, you can Plug the Arduino to your PC via USB cable.

Blink led

Now you're actually ready to "burn" your first program on the arduino board. To select "blink led", the physical translation of the well known programming "hello world", select

File>Sketchbo

ok> Arduino-

0017>Example

s>

Digital>Blink



radiospares

RADIONICS



Once you have your sketch you'll see something very close to the screenshot on the right.

In **Tools>Board** select Now you have to go to

```

Blink | Arduino 0017
File Edit Sketch Tools Help
Blink$
int ledPin = 13; // LED connected to digital pin 13

// The setup() method runs once, when the sketch starts

void setup() {
  // initialize the digital pin as an output:
  pinMode(ledPin, OUTPUT);
}

// the loop() method runs over and over again,
// as long as the Arduino has power

void loop()
{
  digitalWrite(ledPin, HIGH); // set the LED on
  delay(1000); // wait for a second
  digitalWrite(ledPin, LOW); // set the LED off
  delay(1000); // wait for a second
}
  
```

Tools>SerialPort and select the right serial port, the one arduino is attached to.

 Done compiling.
 
 TX RX Flashing
  Blinking Led!

Press Compile button (to check for errors)

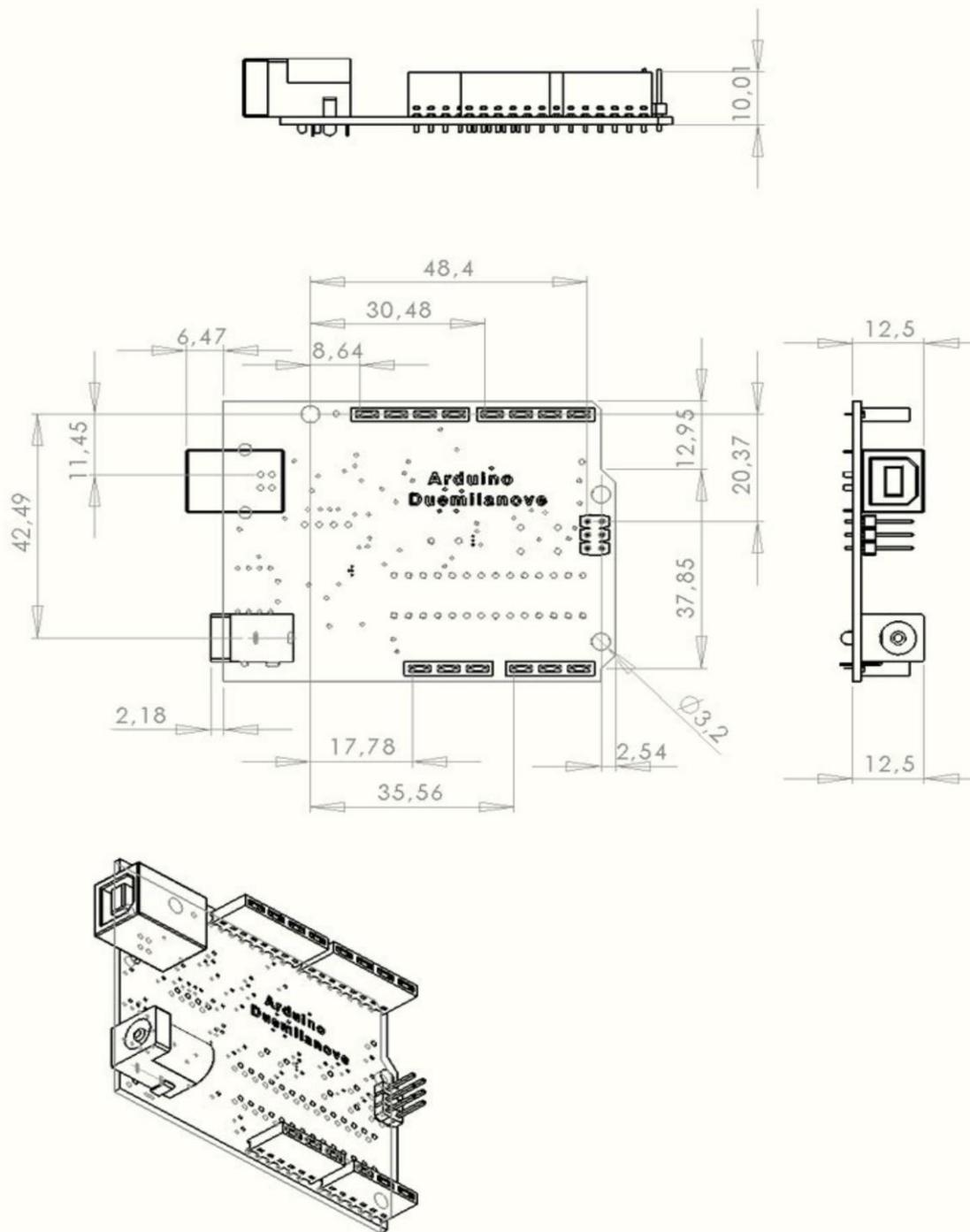


radiospares

RADIONICS



Dimensioned Drawing



radiospares **RADIONICS**



Terms & Conditions



1. Warranties

- 1.1 The producer warrants that its products will conform to the Specifications. This warranty lasts for one (1) years from the date of the sale. The producer shall not be liable for any defects that are caused by neglect, misuse or mistreatment by the Customer, including improper installation or testing, or for any products that have been altered or modified in any way by a Customer. Moreover, The producer shall not be liable for any defects that result from Customer's design, specifications or instructions for such products. Testing and other quality control techniques are used to the extent the producer deems necessary.
- 1.2 If any products fail to conform to the warranty set forth above, the producer's sole liability shall be to replace such products. The producer's liability shall be limited to products that are determined by the producer not to conform to such warranty. If the producer elects to replace such products, the producer shall have a reasonable time to replacements. Replaced products shall be warranted for a new full warranty period.
- 1.3 EXCEPT AS SET FORTH ABOVE, PRODUCTS ARE PROVIDED "AS IS" AND "WITH ALL FAULTS." THE PRODUCER DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE
- 1.4 Customer agrees that prior to using any systems that include the producer products, Customer will test such systems and the functionality of the products as used in such systems. The producer may provide technical, applications or design advice, quality characterization, reliability data or other services. Customer acknowledges and agrees that providing these services shall not expand or otherwise alter the producer's warranties, as set forth above, and no additional obligations or liabilities shall arise from the producer providing such services.
- 1.5 The Arduino™ products are not authorized for use in safety-critical applications where a failure of the product would reasonably be expected to cause severe personal injury or death. Safety-Critical Applications include, without limitation, life support devices and systems, equipment or systems for the operation of nuclear facilities and weapons systems. Arduino™ products are neither designed nor intended for use in military or aerospace applications or environments and for automotive applications or environment. Customer acknowledges and agrees that any such use of Arduino™ products which is solely at the Customer's risk, and that Customer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.
- 1.6 Customer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products and any use of Arduino™ products in Customer's applications, notwithstanding any applications-related information or support that may be provided by the producer.

2. Indemnification

The Customer acknowledges and agrees to defend, indemnify and hold harmless the producer from and against any and all third-party losses, damages, liabilities and expenses it incurs to the extent directly caused by: (i) an actual breach by a Customer of the representation and warranties made under this terms and conditions or (ii) the gross negligence or willful misconduct by the Customer.

3. Consequential Damages Waiver

In no event the producer shall be liable to the Customer or any third parties for any special, collateral, indirect, punitive, incidental, consequential or exemplary damages in connection with or arising out of the products provided hereunder, regardless of whether the producer has been advised of the possibility of such damages. This section will survive the termination of the warranty period.



4. Changes to specifications

The producer may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." The producer reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.



Environmental Policies



The producer of Arduino™ has joined the Impatto Zero® policy of LifeGate.it. For each Arduino board produced is created / looked after half squared Km of Costa Rica's forest's.

Sawing

C7SS

190mm (7-1/2") Circular Saw



Features

- For 190mm (7-1/2") Blade
- Compact and lightweight 3.4kg
- Complied for CE, European standard
- Rigid aluminum base for fine cut
- Side handle easy to grip
- Backward dust emission

Specifications

Capacity: Blade Dia.	190mm (7-1/2")
Max. Cut Depth:90°	68mm (2-11/16")
Max. Cut Depth:45°	46mm (1-7/8")
Power Input	1,050W
No Load Speed	5,500/min.
Overall Length	291mm (11-15/32")
Weight	3.4kg (7.5lbs.)
Standard Accessories	Blade, Guide, Wrench

Note: Manufacturer reserves the right to change specifications of parts and accessories without notice. Specifications and standard accessories may vary from country to country.



UNIVERSITAS BRAWIJAYA

LAMPIRAN III

Listing Program

```
#include <TimerOne.h>

//pin penggerak motor potensio
unsigned int pin_naik=5;
unsigned int pin_turun=6;

//pin counter rpm
unsigned int pin_rpm=2;

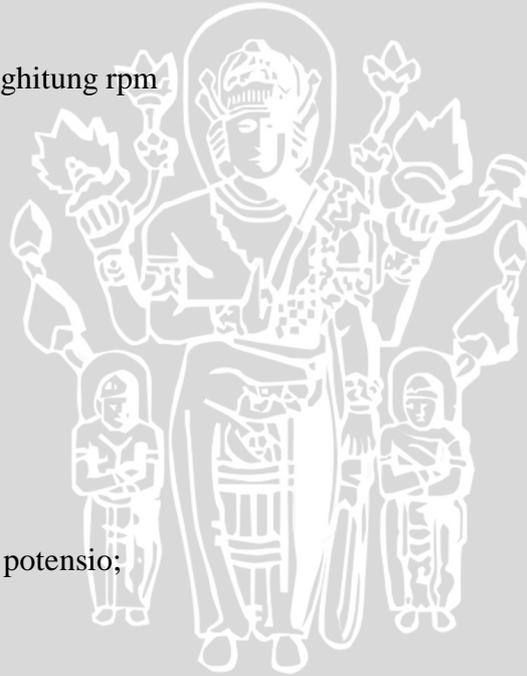
//pin tegangan potensio
unsigned int pin_pot=0;

//variable counter untuk menghitung rpm
unsigned int counter;
unsigned long rpm;
float rotasi;
unsigned long c;
int index = 0;

//variable tampung tegangan potensio;
float v_pot;
//margin v_pot;
float margin=10;

//setpoint tegangan potensio
float SP_Pot = 2550;

//set point rpm
unsigned int SP_rpm = 4500;
```



```
//variable PID
float KP,KI,KD;
float P;
float I;
float D;
float PID_kons;
float Pot_set;
long err;
long lastErr;
long diff;
long errTotal;

//
int i=0;
int i_kirim=0;
int T_sampling = 10;
int T_kirim = 10;

String dataSerial;
void setup(){
  Serial.begin
  (9600);

  //inisialisasi pin counter rpm
  digitalWrite(pin_rpm,HIGH);
```

UNIVERSITAS BRAWIJAYA



```
attachInterrupt(0,counting,RISING);
```

```
//inisialisasi timer untuk sampling menghitung rpm
```

```
Timer1.initialize(40000);//sampling 40ms
```

```
Timer1.attachInterrupt( timerIsr );
```

```
//init potensio
```

```
initPotensio();
```

```
//inisialisasi PID
```

```
KP=0;
```

```
KI=0;
```

```
KD=0;
```

```
//tunggu 1 detik
```

```
delay(1000);
```

```
i=0;
```

```
}
```

```
void loop(){
```

```
  //untuk sampling PID
```

```
  if(i>=T_sampling){
```

```
    err = SP_rpm - rpm;
```

```
    diff = err - lastErr;
```

UNIVERSITAS BRAWIJAYA



```
P = KP * err;
```

```
I = KI * errTotal;
```

```
D = KD * diff;
```

```
PID_kons = P+I+D;
```

```
Pot_set = SP_Pot+PID_kons;
```

```
if(Pot_set>=5000) Pot_set = 5000;
```

```
setPotensio(Pot_set);
```

```
//catat error sebelumnya
```

```
lastErr = err;
```

```
errTotal +=err;
```

```
i=0;
```

```
}
```

```
//untuk kirim data
```

```
if(i_kirim>=T_kirim){
```

```
if(rpm<10){
```

```
Serial.print("000");
```

```
}else if(rpm<100){
```

```
Serial.print("00");
```

```
}else if(rpm<1000){
```

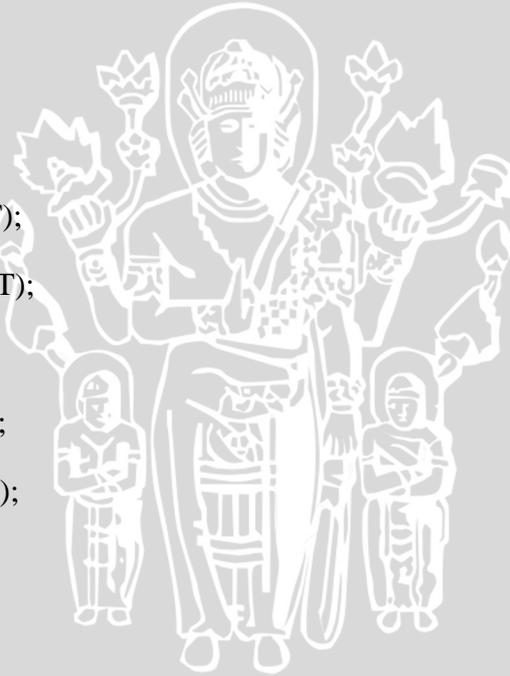
```
Serial.print("0");
```

```
}
```



```
Serial.println(rpm);  
//Serial.println(Pot_set);  
  
i_kirim=0;  
}  
i++;  
i_kirim++;  
delay(1);  
}  
  
void initPotensio(){  
pinMode(pin_naik,OUTPUT);  
pinMode(pin_turun,OUTPUT);  
  
digitalWrite(pin_naik,LOW);  
digitalWrite(pin_turun,LOW);  
}  
  
void setPotensio(float v){  
while(1){  
v_pot = teganganPotensio();  
if(v_pot>(v+margin)){  
digitalWrite(pin_turun,HIGH);  
}  
else if(v_pot<(v-margin)){  
digitalWrite(pin_naik,HIGH);  
}
```

UNIVERSITAS BRAWIJAYA



```
}  
delay(30);  
digitalWrite(pin_naik,LOW);  
digitalWrite(pin_turun,LOW);  
delay(2);  
if(v_pot<=(v+margin) && v_pot>=(v-margin)){  
  break;  
}  
}  
}
```

```
float teganganPotensio(){  
  int d = analogRead(pin_pot);  
  return (float) (d * 4.888);  
}
```

```
void counting(){  
  counter++;  
}
```

```
void timerIsr(){  
  c+=counter;  
  index++;  
  if(index==5){
```

UNIVERSITAS BRAWIJAYA



```
c = c/5;
rotasi = (float)c/24;//(counter * (60000 / Tsample));
rpm = rotasi*1500;
//Serial.println(c);
c=0;
index=0;
}
counter=0;
}
/*
void serialEvent(){
String str_kp="";
String str_ki="";
String str_kd="";

int index=0;
char floatbuf[8];
while(Serial.available() > 0){
char inChar = (char)Serial.read();
// add it to the inputString:
if (inChar == '\n') {
str_kp.toCharArray(floatbuf,sizeof(floatbuf));
KP = atof(floatbuf);
```

UNIVERSITAS BRAWIJAYA



```
str_ki.toCharArray(floatbuf,sizeof(floatbuf));  
KI = atof(floatbuf);  
  
str_kd.toCharArray(floatbuf,sizeof(floatbuf));  
KD = atof(floatbuf);  
  
break;  
}  
if(inChar == ';'){  
    index++;  
    continue;  
}  
if(index==0){  
    str_kp += inChar;  
}else if(index==1){  
    str_ki += inChar;  
}else if(index==2){  
    str_kd += inChar;  
}  
}  
}  
*/
```

UNIVERSITAS BRAWIJAYA

