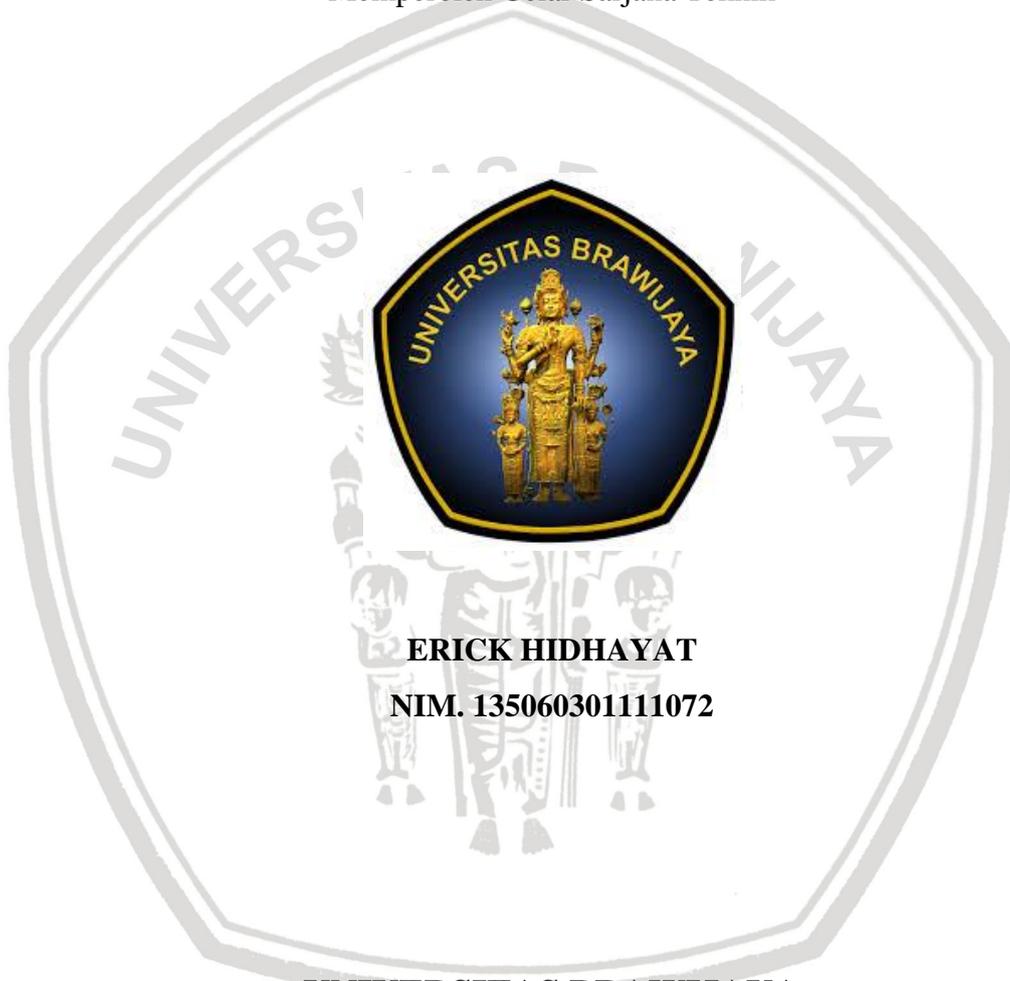


**SISTEM KONTROL KECEPATAN PEMINTALAN BENANG
PADA SUMBU KEMBANG API BERBASIS ARDUINO**

SKRIPSI

TEKNIK ELEKTRO KONSENTRASI TEKNIK KONTROL

Diajukan Untuk Memenuhi Persyaratan
Memperoleh Gelar Sarjana Teknik



**ERICK HIDHAYAT
NIM. 135060301111072**

UNIVERSITAS BRAWIJAYA

FAKULTAS TEKNIK

MALANG

2018

LEMBAR PENGESAHAN
SISTEM KONTROL KECEPATAN PEMINTALAN BENANG PADA
SUMBU KEMBANG API BERBASIS ARDUINO

SKRIPSI
TEKNIK ELEKTRO KONSENTRASI TEKNIK KONTROL

Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik



ERICK HIDHAYAT
NIM. 135060301111072

Skripsi ini telah direvisi dan disetujui oleh dosen pembimbing
pada tanggal 19 Desember 2018

Mengetahui,

Ketua Jurusan Teknik Elektro


Ir. Hadi Sutono, S.T., M.T., Ph.D. IPM
NIP. 19730520 200801 1 013

Dosen Pembimbing


Dr. Ir. Bambang Siswojo, M.T.
NIP. 19621211 198802 1 001

JUDUL SKRIPSI:

SISTEM KONTROL KECEPATAN PEMINTALAN BENANG PADA SUMBU
KEMBANG API BERBASIS ARDUINO

Nama Mahasiswa : ERICK HIDHAYAT

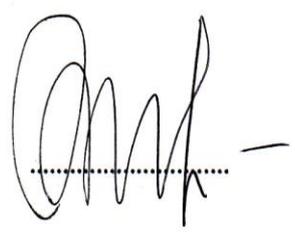
NIM : 135060301111072

Program Studi : TEKNIK ELEKTRO

Konsentrasi : TEKNIK KONTROL

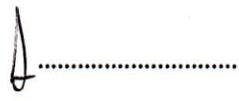
Komisi Pembimbing :

Ketua : Dr. Ir. Bambang Siswojo, M.T.



Tim Dosen Penguji :

Dosen Penguji 1 : Goegoes Dwi Nusantoro, ST., MT.



Dosen Penguji 2 : Dipl.Ing Ir. Moch. Rusli



Dosen Penguji 3 : Rahmadwati, ST., MT., Ph.D



Tanggal Ujian : 19 Desember 2018

SK Penguji : No. 2879 Tahun 2018





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 di 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 dibatalkan, serta diproses sesuai dengan peraturan perundang - undangan yang berlaku (UU No. 20 Tahun 2003, pasal 25 ayat 2 dan pasal 70).

Malang, 19 Desember 2018

Mahasiswa,



ERICK HIDHAYAT

NIM. 135060301111072

RINGKASAN

Erick Hidhayat, Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya, Desember 2018, Sistem Kontrol Kecepatan Pemintalan Benang Pada Sumbu Kembang Api Berbasis Arduino, Dosen Pembimbing: Bambang Siswojo.

Pengaplikasian kembang api di era modern seperti saat ini memiliki kegunaan dan fungsi berbeda tergantung penggunaan dan pemanfaatannya, sebagian besar penggunaan kembang api tersebut sebatas pada penggunaan untuk perayaan tahun baru, euphoria bulan Ramadhan, perayaan Natal dan perayaan hari besar keagamaan lainnya. Terdapat macam – macam jenis sumbu kembang api, yaitu terbuat dari kertas, tissue dan yang banyak beredar dipasaran adalah terbuat dari benang. Akan tetapi belum adanya alat atau pembuatan sumbu kembang api di Indonesia yang diproduksi secara modern. Oleh karena itu perlu diciptakan teknologi yang dapat dimanfaatkan oleh pabrik untuk pembuatan sumbu kembang api baik skala kecil maupun skala besar. Selanjutnya penelitian dan pengujian akan difokuskan pada tahap 1 (satu) yaitu proses pemintalan benang sumbu kembang api. Alat ini menggunakan kontroler Arduino Uno serta motor DC. Arduino Uno adalah sebuah rangkaian elektronik yang dapat mengerjakan berbagai fungsi-fungsi kontrol pada level-level yang kompleks. Arduino Uno dapat diprogram, dikontrol, dan dioperasikan oleh operator yang tidak berpengalaman dalam mengoperasikan komputer. Berdasarkan data respons sistem yang diperoleh dari pengujian dengan menggunakan metode 1 Ziegler-Nichols, maka parameter kontroler PID dapat ditentukan dengan gain $K_p = 3,3$: $K_i = 8,25$ dan $K_d = 0,66$. Pada pengujian sistem keseluruhan dengan memberikan gangguan didapatkan nilai *time overshoot* 0,8 s, waktu saat *steady state* yaitu 1,25 s, *overshoot* sebesar 9,52%, dan *error* 0,95%. Secara keseluruhan sistem dapat kembali pada keadaan *steady* dan mampu memberikan respon sistem yang baik ketika terjadi gangguan dengan *recovery time* sebesar 0,18 s.

Kata kunci: Arduino Mega 2560, PID, motor dc, *sensor rotary encoder*.

SUMMARY

Erick Hidhayat, Department of Electrical Engineering, Faculty of Engineering, University of Brawijaya, December 2018, The Yarn Spinning Speed Control System on the Arduino-Based Firework Axis, Academic Supervisor: Bambang Siswojo.

The application of fireworks in the modern era like today has different uses and functions depending on their use and utilization, most of the use of fireworks is limited to the use for new year celebrations, euphoria of Ramadan, Christmas celebrations and other religious holidays. There are various types of firework axes, which are made of paper, tissue and many of which are circulating in the market are made of yarn. However, the absence of tools or manufacturing of fireworks in Indonesia is produced in a modern way. Therefore it is necessary to create technology that can be utilized by the factory to manufacture fireworks axes both small and large scale. Furthermore, research and testing will be focused on stage 1 (one), which is the fireworks axis spinning process. This tool uses the Arduino Uno controller and DC motor. Arduino Uno is an electronic circuit that can perform various control functions at complex levels. Arduino Uno can be programmed, controlled and operated by operators who are not experienced in operating computers. Based on system response data obtained from testing using Ziegler-Nichols method 1, the PID controller parameters can be determined with gain $K_p = 3.3$; $K_i = 8.25$ and $K_d = 0.66$. In testing the whole system by providing interference, the time overshoot value was 0.8 s, the time at steady state was 1.25 s, overshoot was 9.52%, and the error was 0.95%. Overall the system can return to a steady state and is able to provide a good system response when a disturbance occurs with a recovery time of 0.18 s.

Keywords: Arduino Mega 2560, PID, dc motor, rotary encoder sensor.

PENGANTAR

Segala puji syukur bagi Allah SWT berkat segala limpahan rahmat-Nya, penulisan laporan Skripsi berjudul “ Sistem Kontrol Kecepatan Pemintalan Benang pada Sumbu Kembang Api Berbasis Arduino” dapat diselesaikan dengan baik. Tak lepas shalawat serta salam tercurahkan kepada junjungan kita Nabi Muhammad SAW yang telah menjadi suri tauladan bagi yang mengharapkan rahmat dan hidayah-Nya.

Tak lupa penulis juga mengucapkan banyak terima kasih kepada pihak yang membantu terselesaikannya laporan skripsi ini.

1. Allah SWT yang telah memberikan kelancaran, kemudahan, ilham dan hidayahnya.
2. Sukemi dan Miswati selaku orang tua penulis yang segenap hati mendukung dan mendoakan terselesainya skripsi ini secara moril dan materil.
3. Keluarga tercinta yang segenap hati mendukung dan mendoakan terselesainya skripsi ini.
4. Bapak Ir. Hadi Suyono, S.T., M.T., Ph.D., IPM. selaku Ketua Jurusan Teknik Elektro Universitas Brawijaya.
5. Ibu Ir. Nurussa'adah, M.T. selaku Sekretaris Jurusan Teknik Elektro Universitas Brawijaya.
6. Bapak Ali Mustofa, S.T., M.T. selaku Ketua Program Studi Jurusan Teknik Elektro Universitas Brawijaya.
7. Bapak Ir. Purwanto, M.T. selaku Ketua Kelompok Dosen Konsentrasi Teknik Kontrol Jurusan Teknik Elektro Universitas Brawijaya.
8. Bapak Dr. Ir. Bambang Siswojo, M.T. selaku Dosen Pembimbing atas segala bimbingan, pengarahan, saran, dan kritik yang telah diberikan selama proses pengerjaan skripsi.
9. Teman-teman Spectrum 2013 yang selalu memberikan dorongan semangat kepada penulis.
10. Teman-teman Kontrol 2013 yang selalu memberikan dorongan semangat kepada penulis.

11. Kekasihku yang selalu memberikan dorongan semangat kepada penulis.
12. Semua pihak yang telah memberikan bantuan serta dukungan baik secara langsung maupun tidak langsung atas penyusunan skripsi ini.

Dalam penyusunan skripsi ini, penulis menyadari bahwa skripsi ini belumlah sempurna, karena keterbatasan ilmu dan kendala–kendala lain yang terjadi selama pengerjaan skripsi ini. Oleh karena itu, penulis berharap kritik dan saran untuk penyempurnaan tulisan di masa yang akan datang. Semoga tulisan ini dapat bermanfaat dan dapat digunakan untuk pengembangan lebih lanjut.

Malang, Desember 2018

Penulis



DAFTAR ISI

RINGKASANi

SUMMARYii

PENGANTARiii

DAFTAR ISI.....v

DAFTAR GAMBAR.....vii

DAFTAR TABELviii

DAFTAR LAMPIRANix

BAB I PENDAHULUAN

1.1 Latar Belakang 1

1.2 Rumusan Masalah 3

1.3 Batasan Masalah..... 3

1.4 Tujuan 3

1.5 Manfaat 3

1.6 Sistematika Pembahasan 3

BAB II TINJAUAN PUSTAKA

2.1 *Motor DC* 5

2.2 *Rotary Encoder* 6

2.3 *Arduino Uno*..... 8

2.3.1 *Daya*..... 9

2.3.2 *Memori* 9

2.3.3 *Input dan Output* 9

2.3.4 *Komunikasi*..... 9

2.4 *Pulsa width modulation (PWM)* 10

2.5 *Kontroler* 10

2.5.1 *Kontroler Proposional (P)* 11

2.5.2 *Kontroler Integral (I)* 11

2.5.3 *Kontroler Differensial (D)* 12

2.5.4 *Kontroler Proposional Integral Differensial (PID)* 12

2.6 *Metode Perancangan Kontroler Proposional Integral Differensial (PID)* 13



2.6.1 Menggunakan Metode <i>Ziegler-Nichols</i>	13
2.7 Gearbox Motor DC	17
BAB III METODE PENELITIAN	
3.1 Spesifikasi Alat	18
3.2 Studi Literatur	18
3.3 Perancangan Dan Perealisasian Alat	19
3.3.1 Perancangan Sistem Mekanik	19
3.3.2 Perancangan Sistem Kontrol	19
3.3.3 Perancangan Blok Diagram Sistem Keseluruhan	19
3.4 Perealisasian Alat	20
3.5 Perancangan Kontroler PID	21
3.5.1 Penentuan Nilai Penguatan Kontroler	21
3.6 Perancangan Perangkat Lunak	23
BAB IV HASIL DAN PEMBAHASAN	
4.1 Pengujian Sensor <i>Rotary Encoder</i>	26
4.2 Pengujian Respon Kurva S pada Plant	28
4.3 Pengujian Sistem Secara Keseluruhan	31
BAB V PENUTUP	
5.1 Kesimpulan	36
5.2 Saran	36
DAFTAR PUSTAKA	37
LAMPIRAN	38

DAFTAR GAMBAR

Gambar 2.1	Motor DC.....	6
Gambar 2.2	Sensor <i>Rotary Encoder</i>	7
Gambar 2.3	Linier Driver Output Sensor <i>Rotary Encoder</i>	7
Gambar 2.4	Arduino Mega 2560.....	8
Gambar 2.5	Sinyal PWM Secara Umum.....	10
Gambar 2.6	Diagram Blok Kontroler <i>Proposional</i>	12
Gambar 2.7	Diagram Blok Kontroler <i>Integral</i>	12
Gambar 2.8	Diagram Blok Kontroler <i>Differensial</i>	13
Gambar 2.9	Diagram Blok Kontroler PID.....	13
Gambar 2.10	Kurva Respon <i>Unit Step</i> yang Menunjukkan 25% Maximum Overshoot.....	14
Gambar 2.11	Respon <i>Plant</i> Terhadap Masukkan Berupa <i>Unit Step</i>	14
Gambar 2.12	Kurva Respon yang berbentuk S	15
Gambar 2.13	Sistem Loop Tertutup dengan Kontroler <i>Proposional</i>	16
Gambar 2.14	Osilasi Berkesinambungan dengan Periode <i>Pcr</i>	17
Gambar 2.15	Gearbox Motor DC	17
Gambar 3.1	Perancangan Blok Diagram Sistem Keseluruhan	19
Gambar 3.2	Pemintalan Sumbu Kembang Api	21
Gambar 3.3	Kurva Respon Yang Berbentuk S.....	22
Gambar 3.4	Flowchart Perangkat Lunak	24
Gambar 4.1	Diagram Perancangan Peralatan Pengujian Sensor	26
Gambar 4.2	Respon Pengujian Sensor <i>Rotary Encoder</i>	28
Gambar 4.3	Diagram Perancangan Peralatan Pengujian <i>Plant</i>	29
Gambar 4.4	Grafik Pengujian Kurva S.....	30
Gambar 4.5	Diagram Peralatan Pengujian Sistem Keseluruhan	32
Gambar 4.6	Grafik hasil pengujian untai tertutup dengan setpoint 210 rpm dan memberikan gangguan sementara.....	35

DAFTAR TABEL

Tabel 2.1 Konfigurasi pin dengan warna kabel 7

Tabel 2.2 Aturan penalaan Ziegler-Nichols berdasarkan Respon *Unit Step* dari *Plant*..... 16

Tabel 2.3 Aturan Ziegler-Nichols berdasarkan *Critical Gain Kcr* dan *Critical periode Per* 17

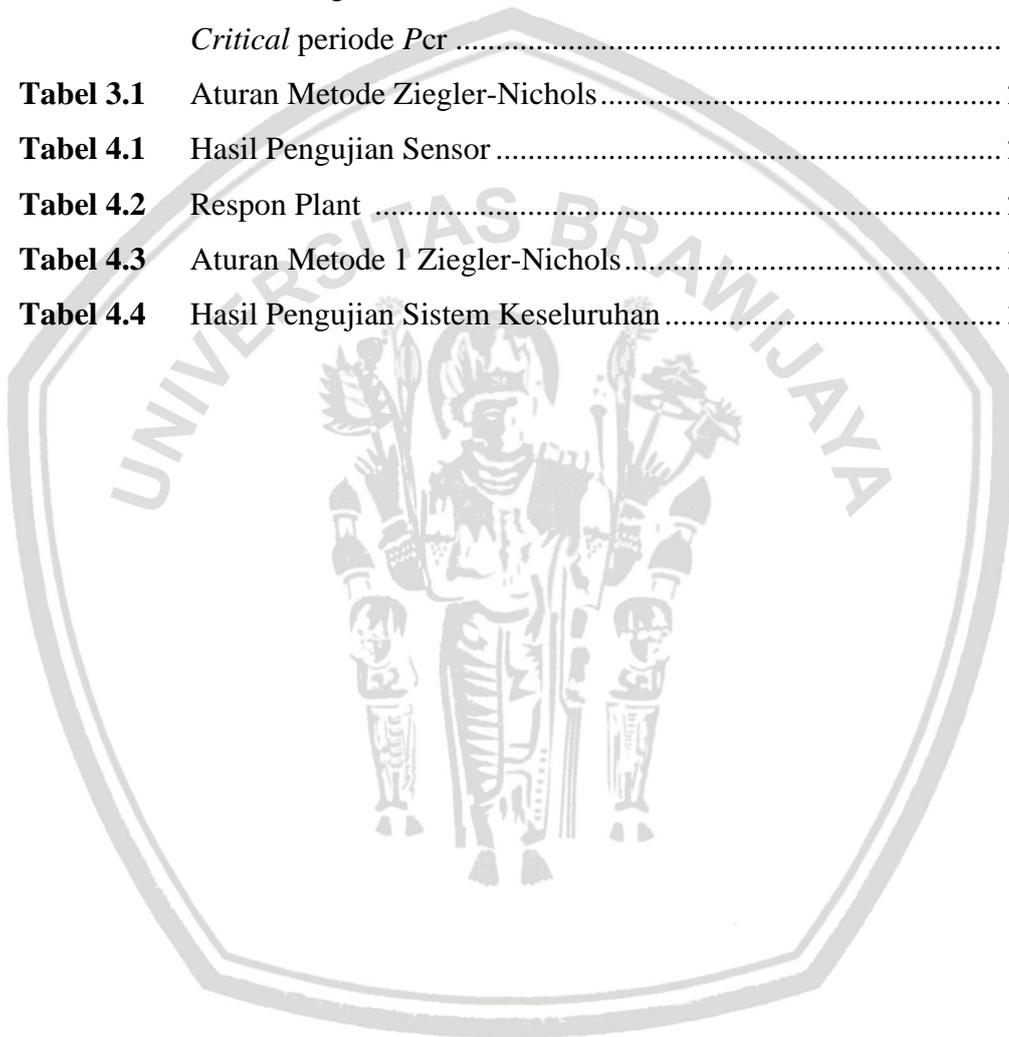
Tabel 3.1 Aturan Metode Ziegler-Nichols..... 23

Tabel 4.1 Hasil Pengujian Sensor 27

Tabel 4.2 Respon Plant 29

Tabel 4.3 Aturan Metode 1 Ziegler-Nichols..... 31

Tabel 4.4 Hasil Pengujian Sistem Keseluruhan..... 32



DAFTAR LAMPIRAN

Lampiran 1 Dokumentasi Alat	38
Lampiran 2 Datasheet Arduino ATmega 2560.....	40



BAB I

PENDAHULUAN

1.1 LATAR BELAKANG

Sejarah petasan bermula dari China sekitar abad ke-9 ketika seorang juru masak secara tak sengaja mencampurkan tiga bahan, yaitu bubuk hitam (black powder) yakni garam peter atau kalium nitrat, belerang (sulfur) dan arang dari kayu (charcoal) yang berasal dari dapurnya. Ternyata campuran ketiga bahan itu mudah terbakar. Jika ketiga bahan tersebut dimasukan ke dalam sepotong bambu yang ada sumbunya yang lalu dibakar dan akan meletus serta mengeluarkan suara ledakan yang dipercaya mampu mengusir roh jahat. Dalam perkembangannya, petasan jenis ini dipercaya dipakai juga dalam perayaan pernikahan, kemenangan perang, peristiwa gerhana bulan dan upacara-upacara keagamaan. Pada saat masa Dinasti Song sekitar 960-1279 Masehi, didirikan pabrik petasan yang kemudian menjadi dasar dari pembuatan kembang api karena lebih menitik-beratkan pada warna-warni dan bentuk pijar api di angkasa. Tradisi petasan lalu menyebar ke seluruh pelosok dunia. Pada masa Renaissance, di Italia dan Jerman ada sekolah yang khusus mengajarkan masalah pembuatan kembang api. Di sekolah Italia menekankan pada kerumitan kembang api, sedangkan di sekolah Jerman menekankan pada kemajuan ilmu pengetahuan. Dan akhirnya muncul istilah Pyrotechnics yang menggambarkan seni membuat kembang api. Untuk membuat kembang api dibutuhkan seorang ahli yang mengerti reaksi fisika dan kimia (Asmarawati, 2014).

Pengaplikasian kembang api di era modern seperti saat ini memiliki kegunaan dan fungsi berbeda tergantung penggunaan dan pemanfaatannya, sebagian besar penggunaan kembang api tersebut sebatas pada penggunaan untuk perayaan tahun baru, euphoria bulan Ramadhan, perayaan Natal dan perayaan hari besar keagamaan lainnya. Pemanfaatan kembang api untuk meminta pertolongan ketika seseorang berada di hutan dengan cara menyalakan kembang api yang berasap dan memiliki warna tertentu, dengan tujuan afar tim SAR atau tim penyelamat mengetahui posisi orang tersebut.

Komponen yang tidak kalah penting dalam kembang api adalah sumbu kembang api. Berbagai jenis sumbu kembang api, ada yang terbuat dari kertas, tissue dan yang banyak beredar dipasaran adalah terbuat dari benang (Haflan, 2008). Akan tetapi belum adanya alat atau pembuatan sumbu kembang api di Indonesia yang diproduksi secara modern.

Berdasarkan permasalahan diatas maka perlu diciptakan teknologi yang dapat dimanfaatkan oleh pabrik untuk pembuatan sumbu kembang api baik skala kecil maupun skala besar di Indonesia. Solusinya adalah teknologi yang mampu meningkatkan pembuatan sumbu kembang api. Alat ini merupakan teknologi yang mampu meningkatkan kualitas dari sumbu kembang api. Dengan terciptanya alat ini kita dapat membuat kembang api secara baik.

Ada 3 (tiga) proses tahapan dari pembuatan sumbu kembang api dari benang tahap 1 (satu) dimulai dari pemintalan benang untuk menjebak bubuk sumbu, setelah itu tahap 2 (dua) dimana benang akan masuk ke dalam sebuah cairan kental untuk melapisi benang dan dilanjutkan ke proses pemanasan untuk mengeringkan cairan kental tersebut, proses ini berfungsi untuk menghindarkan sumbu agar tidak basah terkena air dan juga memberi ketahanan pada sumbu kembang api, lalu dilanjutkan ke tahap 3 (tiga) yaitu proses penggulungan sumbu kembang api dimana sumbu akan ditarik oleh motor stepper dan dikontrol oleh pegas yang sudah diberikan potensiometer sebagai sensor yang akan di sinkronkan dengan aktuator pada proses penggulungan akhir.

Alat ini menggunakan kontroler Arduino Uno. Arduino Uno adalah sebuah rangkaian elektronik yang dapat mengerjakan berbagai fungsi-fungsi kontrol pada level-level yang kompleks. Arduino Uno dapat diprogram, dikontrol, dan dioperasikan oleh operator yang tidak berpengalaman dalam mengoperasikan komputer. Diharapkan dengan menggunakan kontroler Arduino Uno parameter yang diukur dapat dikendalikan dengan baik.

Selanjutnya penelitian dan pengujian akan difokuskan pada tahap 1 (satu) yaitu proses pemintalan benang sumbu kembang api. Alat ini menggunakan kontroler Arduino Uno serta motor DC. Arduino Uno adalah sebuah rangkaian elektronik yang dapat mengerjakan berbagai fungsi-fungsi kontrol

pada level-level yang kompleks. Arduino Uno dapat diprogram, dikontrol, dan dioperasikan oleh operator yang tidak berpengalaman dalam mengoperasikan komputer. Diharapkan dengan menggunakan kontroler Arduino Uno parameter yang diukur dapat dikendalikan dengan baik.

1.2 RUMUSAN MASALAH

1. Bagaimana merancang sistem kontrol kecepatan pemintalan benang?
2. Bagaimana respon sistem kontrol kecepatan pemintalan benang berbasis Arduino?

1.3 BATASAN MASALAH

Dengan mengacu pada permasalahan yang telah dirumuskan, maka hal-hal yang berkaitan dengan alat diberi batasan sebagai berikut :

1. Perancangan dan pembuatan alat sistem kontrol kecepatan pemintalan benang adalah desain sendiri.
2. Pembahasan ditekankan pada pengendalian motor untuk menggerakkan plat pada sistem dengan menggunakan Kontrol PID berbasis Mikrokontroler Arduino Uno.

1.4 TUJUAN

Tujuan skripsi ini adalah merancang dan membuat suatu alat pemintalan benang kembang api menggunakan kontroler PID berbasis Arduino Uno.

1.5 MANFAAT

Manfaat skripsi ini adalah dapat mengetahui cara mengontrol pemintalan benang sumbu kembang api menggunakan mikrokontroler PID berbasis Arduino Uno.

1.6 SISTEMATIKA PEMBAHASAN

Skripsi ini terdiri dari 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 Penelitian

Membahas metode penelitian dan perencanaan alat.

BAB IV Perancangan dan Pembuatan Alat

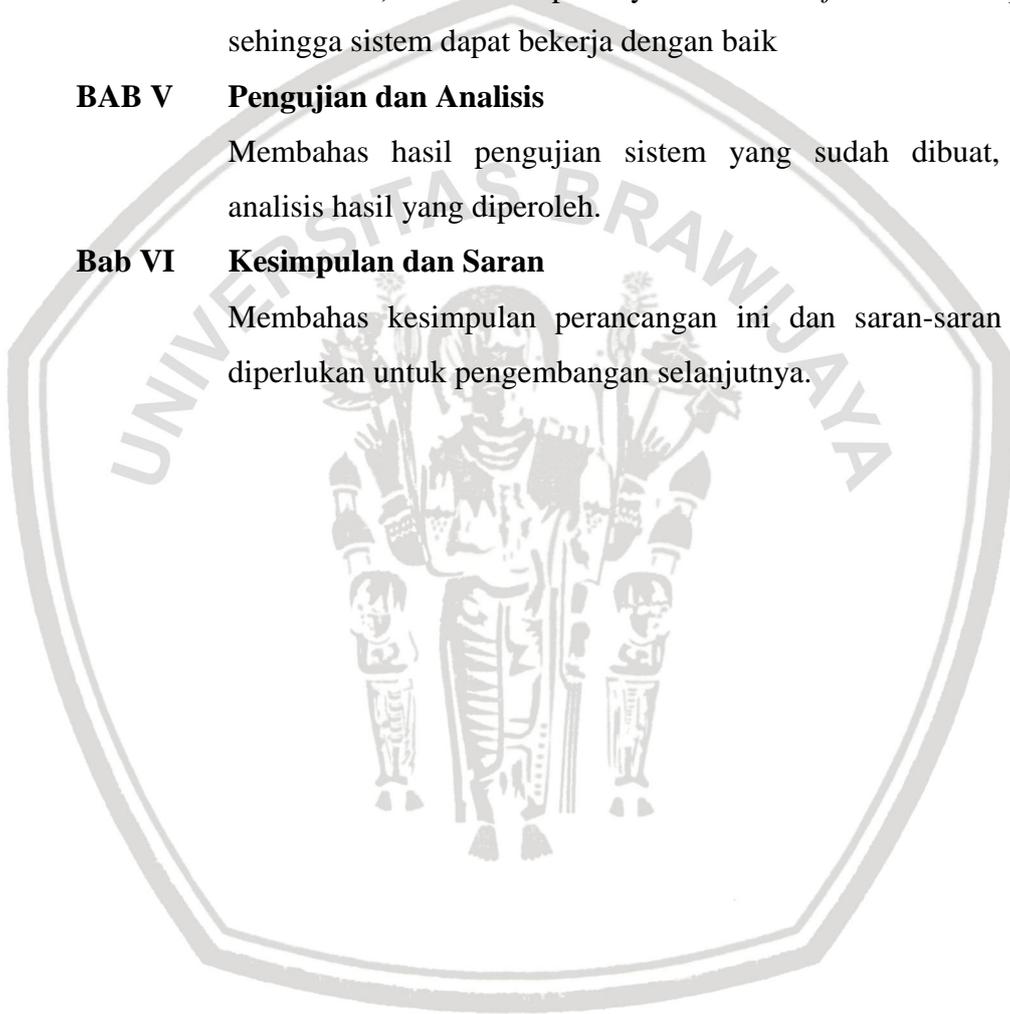
Membahas perancangan kontroler PID (Propositional Integral Differensial) dan menerapkannya ke dalam *software* di komputer, sehingga sistem dapat bekerja dengan baik

BAB V Pengujian dan Analisis

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

Bab VI Kesimpulan dan Saran

Membahas kesimpulan perancangan ini dan saran-saran yang diperlukan untuk pengembangan selanjutnya.



BAB II

TINJAUAN PUSTAKA

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

- Motor Direct Current (DC)
- *Rotary encoder*
- Arduino Mega 2560
- *Pulse Width Modulation (PWM)*
- Kontroler *Proporsional Integral Differensial (PID)*

2.1 Motor DC

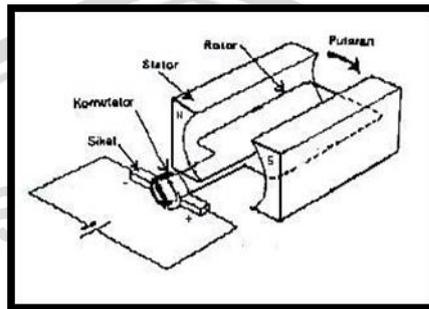
Motor DC yang digunakan dalam penelitian adalah motor DC yang menggunakan permanen magnet. Alasan pemilihan motor DC tipe ini adalah kemudahan dalam pengontrolan dengan menggunakan pengaturan tegangan DC. Medan stator motor jenis ini dihasilkan oleh magnet permanen bukan elektromagnet. motor DC mempunyai kurva kecepatan torsi yang linier dalam jangka yang lebar. Penggunaan magnet permanen tidak membutuhkan daya listrik untuk menghasilkan medan stator, sehingga daya dan pendinginan yang diperlukan lebih rendah dibandingkan motor yang menggunakan elektromagnet. Perubahan kecepatan motor dapat diatur dengan cara mengubah-ubah besarnya tegangan DC yang diberikan. Motor DC memiliki beberapa keunggulan, yaitu (Kadir, 2016):

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

Motor DC hampir sama konstruksinya dengan motor AC, perbedaannya terletak pada sikat dan cincin belah (komutator). Saat siklus pertama, arus mengalir dari kutub positif ke negatif. Aliran arus yang melewati bagian kabel yang berada didekat kutub N magnet akan menimbulkan gaya *Lorentz*

ke bawah. Sementara itu aliran arus yang melewati kabel yang berada di dekat kutub S magnet akan menyebabkan gaya Lorentz ke atas.

Kedua perpaduan gaya Lorentz tersebut akan menyebabkan kawat berputar. Pada siklus berikutnya terjadi hal yang serupa seperti pada siklus sebelumnya. Apabila arus terus-menerus dialirkan, maka kawat akan berputar secara terus menerus pula. Pada aplikasi sesungguhnya, kawat adalah sebuah rotor yang akan dikopel dengan sebuah as dan akan memutar as tersebut terus menerus seiring perputaran motor. Motor DC ditunjukkan dalam Gambar 2.1.



Gambar 2.1 Motor DC

2.2 Rotary Encoder

Sensor *rotary encoder* yang digunakan pada penulisan ini merupakan sensor yang diproduksi oleh Autonics dengan tipe E40H8-500P/R-6-L-5 seperti pada Gambar 2.2. Sensor ini memiliki lubang berukuran 80mm sebagai tempat untuk menghubungkan antara sensor dengan roda. Keluaran dari sensor ini berupa pulsa dengan tegangan $5\text{v} \pm 5\%$ saat pulsa *high* dan 0v saat pulsa *low*. Sensor ini dapat menyediakan 500 pulsa /rotasi. Pulsa keluaran dari sensor ini dapat didefinisikan menjadi gerakan, posisi dan arah. Sehingga posisi sudut suatu poros benda berputar dapat diolah menjadi informasi berupa kode digital oleh *rotary encoder* untuk diteruskan oleh sistem rangkaian kendali (Platt, 2013).



Gambar 2.2 Rotary Encoder

Sumber: autonics 2013

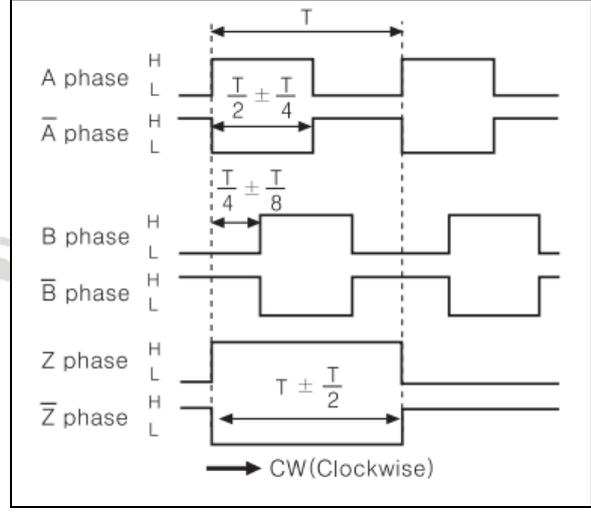
Sensor ini memiliki 9 kaki atau pin. Konfigurasi dari masing-masing pin ditunjukkan pada Tabel 2.1

Tabel 2.1 Konfigurasi pin dengan warna kabel

No pin	Fungsi	Warna Kabel
1	OUT A	Hitam
2	OUT A'	Merah
3	+ V	Coklat
4	GND	Biru
5	OUT B	Putih
6	OUT B'	Abu-abu
7	OUT Z	Jingga
8	OUT Z'	Kuning
9	F.G	Pelindung kabel

sumber: *Datasheet Autonics Rotary Encoder*

Perbedaan antara masing-masing keluaran dari sensor *rotary encoder* ini dapat kita lihat pada **Gambar 2.3**



Gambar 2.3 Line Driver Output Sensor Rotary Encoder

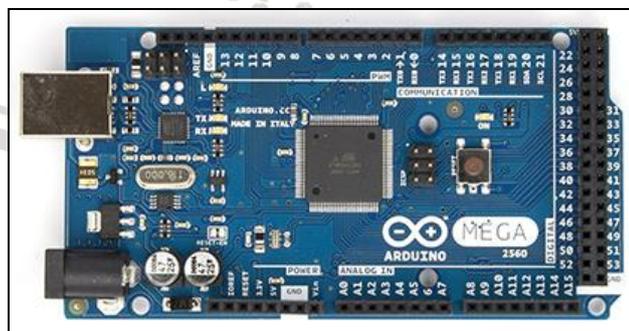


Pada **Gambar 2.3** dapat diketahui bahwa terdapat pergeseran fasa antara OUT A dan OUT B dengan periode pulsa yang sama, sedangkan OUT Z memiliki periode 2 kali lebih lama. Fasa yang berkebalikan juga disediakan oleh rotaru ini pada OUT A', OUT B' dan OUT Z'.

Rotary encoder adalah device elektromekanik yang dapat memonitor gerakan dan posisi. Rotary encoder umumnya menggunakan sensor optik untuk menghasilkan serial pulsa yang dapat diartikan menjadi gerakan, posisi, dan arah. Sehingga posisi sudut suatu dan Gambar 2.5 adalah diagram proses dan diagram blok perancangan sistem kontrol kaskade suhu dan *flow* pada proses pembakaran. poros benda berputar dapat diolah menjadi informasi berupa kode digital oleh rotary encoder untuk diteruskan oleh rangkaian kendali. Rotary encoder dibedakan menjadi dua jenis yakni rotary encoder absolut dan relatif.

2.3 Arduino Uno

Arduino mega 2560 adalah board arduino Mega versi terbaru, yang merupakan perbaikan dari board terdahulunya yaitu arduino Mega 1280. Arduino mega 2560 memiliki 54 digital input / output (dengan 14 pin yang dapat digunakan sebagai output PWM (Pulse Width Modulation)), 16 input analog, 4 UART (Universal Asynchronous Receiver / Transmitter), kristal 16 MHz osilator, koneksi USB, jack listrik, header ICSP (in-circuit serial programming), dan tombol reset. Bentuk Arduino mega 2560 ditunjukkan pada **Gambar 2.4**.



Gambar 2.4 Arduino Mega 2560 Tampak Atas

Sumber: Arduino Mega 2560 Datasheet

2.3.1 Daya

Arduino mega dapat diaktifkan melalui koneksi USB atau dengan catu daya eksternal. Sumber daya akan dipilih secara otomatis. Eksternal (non-USB) daya dapat berasal baik dari adaptor AC-DC atau baterai. Adaptor ini dapat dihubungkan dengan menancapkan sebuah 2.1 mm konektor pusat-positif ke colokan listrik board.

Modul dapat beroperasi pada pasokan eksternal 6 sampai 20 volt. Jika disertakan dengan kurang dari 7V, pin 5V dapat menyediakan kurang dari lima volt dan modul mungkin tidak stabil. Jika menggunakan lebih dari 12V, regulator tegangan bisa panas dan merusak modul. Kisaran yang disarankan adalah 7 sampai 12 volt. lebih mudah untuk menggunakan seperangkat persamaan differensial orde pertama (Ogata, 1997).

2.3.2 Memori

Atmega 2560 memiliki 256 KB memori flash untuk menyimpan kode dengan 8 KB digunakan untuk bootloader, 8 KB SRAM (Static Random Access Memory) dan 4 KB EEPROM (Electrically Erasable Programmable Read-Only Memory).

2.3.3 Input Dan Output

Masing-masing dari 54 digital pin pada Mega dapat digunakan sebagai input atau output, menggunakan `pinMode ()`, `digitalWrite ()`, dan `digitalRead ()`. Dengan 14 pin yang dapat digunakan sebagai output PWM (Pulse Width Modulation) menyediakan 8-bit PWM output dengan fungsi `analogWrite ()`.

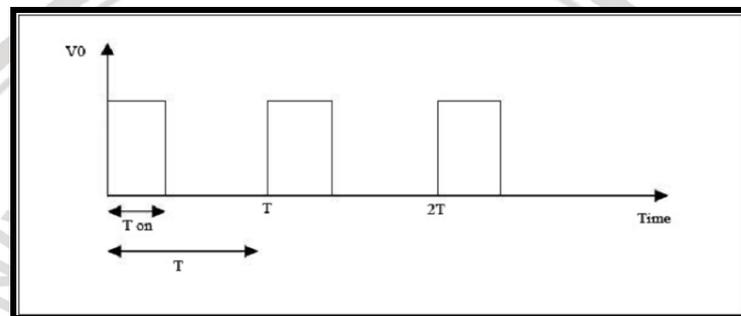
2.3.4 Komunikasi

Arduino Mega2560 memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, Arduino lain, atau mikrokontroler lainnya. ATmega2560 ini menyediakan empat UART hardware untuk komunikasi serial. LED (Light-Emitting Diode) akan berkedip ketika data sedang dikirim melalui ATmega8U2/ATmega16U2 koneksi Chip dan USB ke komputer.

2.4 Pulse Width Modulation (PWM)

Pulse Width Modulation (PWM) digunakan untuk mengatur kecepatan dari motor DC. Kecepatan motor DC tergantung pada besarnya *duty cycle* yang diberikan pada motor DC tersebut. *Duty cycle* adalah rasio atau perbandingan lama waktu suatu sinyal berada dalam kondisi *high* dengan lama waktu suatu sinyal tersebut dalam kondisi (*high+low*).

Pada sinyal PWM, frekuensi sinyal konstan sedangkan *duty cycle* bervariasi dari 0%-100%. Dengan mengatur *duty cycle* akan diperoleh keluaran yang diinginkan. Sinyal PWM secara umum dapat dilihat dalam **Gambar 2.5**.



Gambar 2.5 Sinyal PWM Secara Umum

Sumber: www.electronics-scheme.com

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

Dengan :

T_{on} = Periode logika tinggi

T = Periode keseluruhan

2.5 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* berosilasi terus menerus ketika dikenai suatu gangguan. 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*.

2.5.1 Kontroler Proporsional (P)

Kontroler proporsional adalah sebuah kontroler yang memiliki karakteristik mempercepat respon. Hubungan antara keluaran kontroler $m(t)$ dan sinyal kesalahan penggerak $e(t)$ ditunjukkan pada persamaan 2.2:

2.2:

$$m(t) = K_p e(t) \quad (2.2)$$

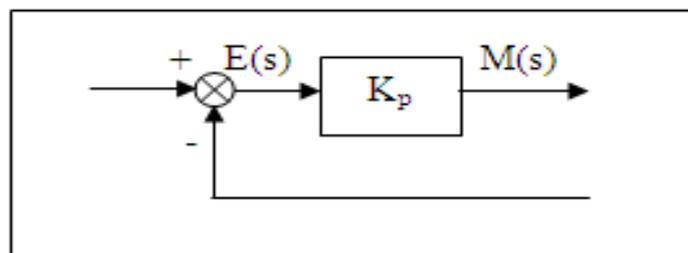
atau, dalam besaran transformasi Laplace ditunjukkan pada persamaan 2.3:

2.3:

$$\frac{M(s)}{E(s)} = K_p \quad (2.3)$$

Dimana 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 pada **Gambar 2.6**.



Gambar 2.6 Diagram Blok Kontroler Proporsional (Ogata K., 1997)

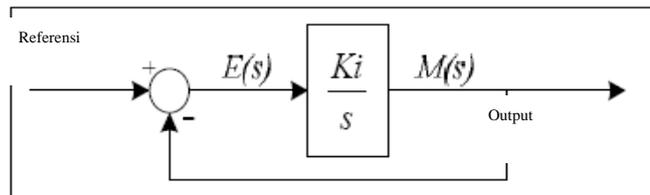
2.5.2 Kontroler Integral (I)

Kontroler integral memiliki kemampuan untuk mengurangi *offset* yang ditinggalkan oleh kontroler proporsional. Harga keluaran kontroler $m(t)$ diubah dengan laju yang sebanding dengan sinyal

kesalahan penggerak $e(t)$. Persamaan kontroler Ki ditunjukkan dalam persamaan 2.4.(Ogata K.,1997).

$$\frac{dm(t)}{dt} = K_i e(t) \quad (2.4)$$

Ki adalah penguatan integral. **Gambar 2.7** menunjukkan diagram blok kontroler integral.



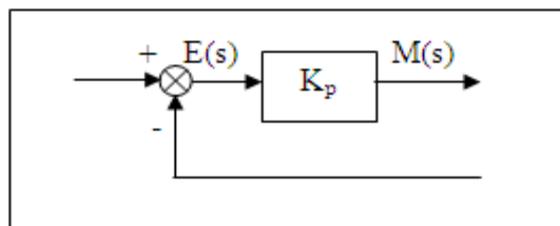
Gambar 2.7 Diagram Blok Kontroler Integral (Ogata K., 1997)

2.5.3 Kontroler Differensial (D)

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 differensial 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 $e(t)$ dinyatakan pada persamaan 2.5 :

$$\frac{M(s)}{E(s)} = K_d \cdot s \quad (2.5)$$

Kd adalah penguatan differensial. **Gambar 2.8** menunjukkan diagram blok kontroler differensial.



Gambar 2.8 Diagram Blok Kontroler Differensial (OgataK., 1997)

2.5.4 Kontroler Proporsional Integral Differensial (PID)

Gabungan aksi kontrol proporsional, integral, dan differensial mempunyai keunggulan dibandingkan dengan masing-masing dari tiga

aksi kontrol tersebut. Masing-masing kontroler P, I, maupun D berfungsi untuk mempercepat reaksi sistem, menghilangkan *offset*, dan mendapatkan energi ekstra ketika terjadi perubahan *load*.

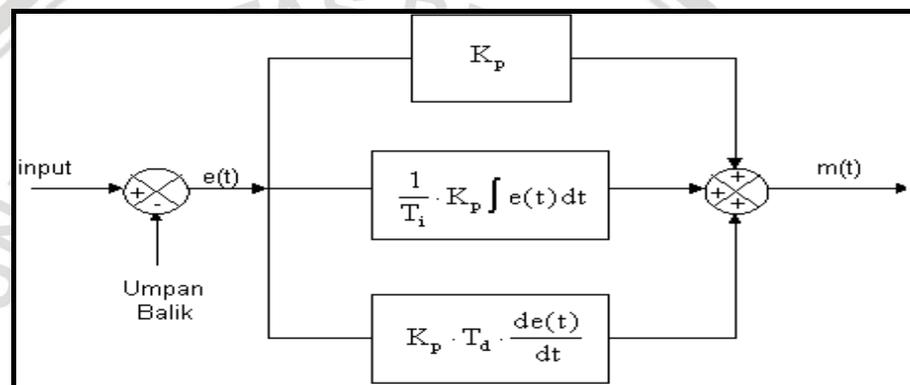
Persamaan kontroler PID ini dapat dinyatakan dalam persamaan 2.6 :

$$m(t) = K_p \cdot e(t) + \frac{K_p}{T_i} \cdot e(t)dt + K_p \cdot T_d \frac{de(t)}{dt} \quad (2.6)$$

Dalam transformasi Laplace dinyatakan dalam persamaan 2.7 :

$$\frac{M(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i s} + T_d \cdot s \right) \quad (2.7)$$

T_i adalah waktu integral dan T_d adalah waktu derivatif. **Gambar 2.9** menunjukkan diagram blok kontroler PID.



Gambar 2.9 Diagram Blok Kontroler PID (OgataK., 1997)

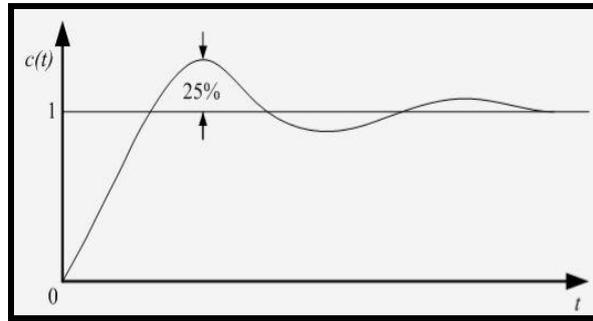
2.6 Metode Perancangan Kontroler *Proporsional Integral Differensial* (PID)

2.6.1 Menggunakan Metode *Ziegler-Nichols*

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

Terdapat dua metode yang disebut dengan aturan penalaan *Ziegler-Nichols*, pada kedua metode tersebut memiliki tujuan yang sama yaitu

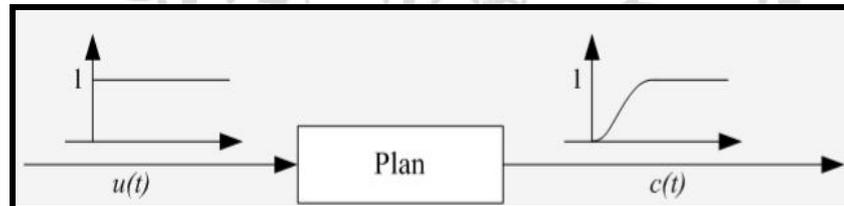
untuk mencapai 25% *maximum overshoot* pada respon *unit step*, seperti ditunjukkan dalam Gambar 2.10.



Gambar 2.10 Kurva Respon *Unit Step* yang Menunjukkan 25% *Maximum Overshoot* (Sumber: Ogata, K., 1997)

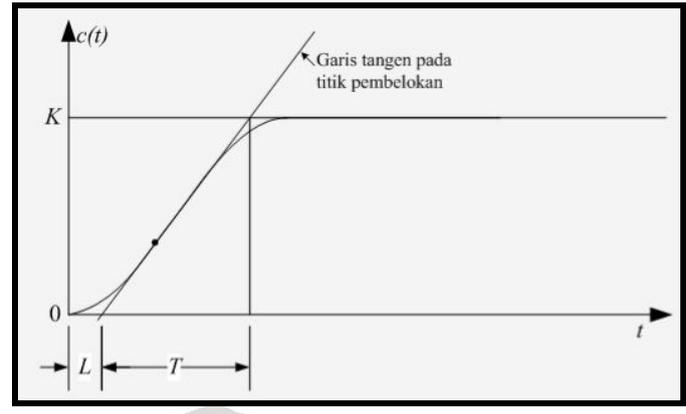
a). Metode Pertama

Metode pertama atau sering disebut metode kurva reaksi, respon dari plan dapat diperoleh secara eksperimental dengan masukan berupa *unit step*, seperti yang ditunjukkan dalam **Gambar 2.11**.



Gambar 2.11 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 dalam Gambar 2.12, jika respon tidak memberikan bentuk kurva S, maka metode ini tidak berlaku. (Ogata, K., 1997).



Gambar 2.12 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 Gambar 2.12. Fungsi alih $C(s)/U(s)$ dapat dilakukan pendekatan dengan sistem orde satu dengan persamaan sebagai berikut:

$$\frac{C(s)}{U(s)} = \frac{K e^{-Ls}}{Ts + 1} \tag{2.8}$$

Ziegler dan Nichols menyarankan untuk menentukan nilai-nilai dari K_p , T_i dan T_d berdasarkan pada formula yang ditunjukkan dalam Tabel 2.2 (Ogata, K., 1997)

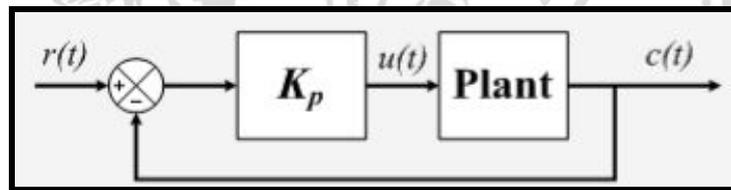
Tabel 2.2 Aturan Penalaan Ziegler-Nichols Berdasarkan Respon Unit Step Dari Plan

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	$1,2 \frac{T}{L}$	$2L$	$0,5 L$



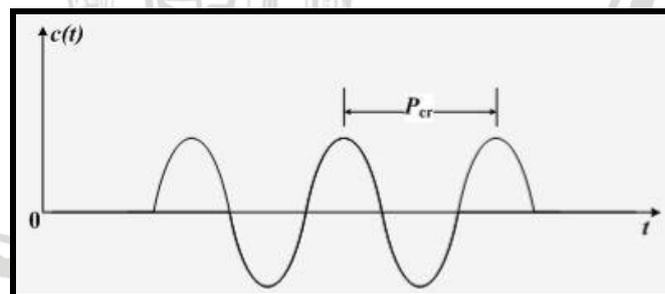
b). Metode Kedua

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* dapat dilihat dalam Gambar 2.13 dan untuk osilasi berkesinambungan dengan periode P_{cr} dapat dilihat dalam Gambar 2.14. Ziegler dan Nichols menyarankan penyetelan nilai parameter K_p, T_i, T_d dan berdasarkan rumus yang diperlihatkan dalam Tabel 2.3. (Ogata, K., 1997).



Gambar 2.13 Sistem Loop Tertutup dengan Kontroler *Proporsional*

(Sumber: Ogata, K., 1997)



Gambar 2.14 Osilasi Berkesinambungan dengan Periode P_{cr}

(Sumber : Ogata, K., 1997)

Tabel 2.3 Aturan Dasar Ziegler-Nichols Berdasarkan Critical Gain Kcr Dan Critical Period Pcr

Tipe Kontroler	K_p	T_i	T_d
P	0.5 Kcr	∞	0
PI	0.45 Kcr	$\frac{1}{1.2} Pcr$	0
PID	0.60 Kcr	0.5 Pcr	0.125 Pcr

(Sumber: Ogata, K., 1997)

2.7 Gearbox Motor DC

Motor dicatu dengan tegangan 3-6 Volt DC dengan kecepatan putar sumbu 12300 rpm pada tegangan 3 Volt. Cocok untuk aplikasi robotik yang memerlukan kecepatan dan kekuatan. Juga tersedia sejumlah pilihan roda yang dapat dengan mudah dihubungkan dengan poros gearbox yang ada.

Selain seri di atas, motor DC yang telah dilengkapi dengan gearbox-KR300, yang memberikan output 300 rpm dan torsi 2.4 kg.cm pada tegangan catu 5 Volt DC. Memiliki bentuk fisik dan dimensi seperti servo, sehingga cukup mudah untuk pemasangannya.



Gambar 2.14 Gearbox Motor DC

Motor DC kebanyakan memiliki bentuk fisik bulat, sehingga tidak mudah untuk memasangnya pada chassis. Kecepatan putarnya (RPM) tinggi, namun torsinya rendah sehingga perlu dilengkapi dengan gearbox.

BAB III METODE PENELITIAN

Dalam skripsi ini merancang sebuah alat uji coba secara statis yaitu sistem kontrol kecepatan yang diuji berada dalam keadaan diam, yang terdiri dari Pemintalan sumbu kembang api untuk menggerakkan 2 motor dc dan pada pemintalan dibuat gangguan berupa perubahan ketegangan pemintalan sumbu kembang api. Alat uji ini adalah sebuah alat yang mampu mengendalikan kecepatan agar stabil sumbu kembang api pada proses pemintalan.

Dalam bab ini membahas mengenai perancangan dan pembuatan sistem pengendali kecepatan pada motor dc dengan menggunakan Kontroler Proporsional Integral Differensial (PID). Perancangan perangkat tersebut meliputi perangkat keras maupun perancangan perangkat lunak. Sedangkan pembuatan ini bertujuan untuk menghasilkan semua perangkat pendukung maupun alat secara keseluruhan.

3.1 Spesifikasi Alat

Spesifikasi alat secara keseluruhan harus ditentukan terlebih dahulu agar didapatkan kinerja sistem yang efektif dan sesuai yang diinginkan. Spesifikasi alat yang akan direalisasikan sebagai berikut :

1. Alat kecepatan pemintalan sumbu kembang api yang digunakan berbahan alumunium
2. Mikrokontroler yang digunakan adalah arduino uno
3. Kontroler yang digunakan adalah PID
4. *Software* yang digunakan untuk pemrograman Arduino 1.8.2

3.2 Studi Literatur

Studi literatur mengacu pada prinsip kerja alat yang dirancang meliputi sistem pengendalian motor pemintalan benang sumbu kembang api menggunakan kontroler PID, karakteristik dari komponen yang digunakan serta studi tentang pengendalinya yaitu menggunakan arduino uno sebagai pusat pengontrol dengan kendali kontroler PID.

3.3 Perancangan Dan Perealisasian Alat

3.3.1 Perancangan Sistem Mekanik

Sebelum mendesain alat pemintalan sumbu kembang api secara keseluruhan diperlukan alat uji untuk mengontrol kecepatan sumbu kembang api. Alat bantu desain sistem pemintalan ini menggunakan 1 aktuator motor dc, dan sensor *rotary encoder*. Sebagai gangguan adalah ketegangan benang saat pemintalan yang cenderung berubah-ubah. Dengan menggunakan kontroler PID, dapat diuji ketegangan benang yang dihasilkan. Sehingga kecepatan putaran pemintalan dengan ketegangan sumbu kembang api dapat distabilkan dengan kontroler PID.

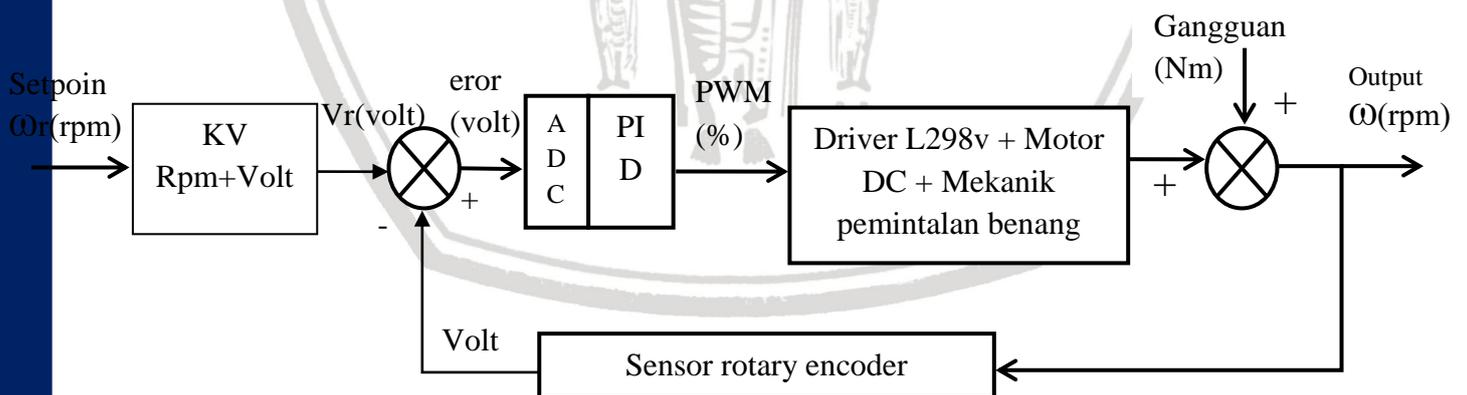
3.3.2 Perencanaan Sistem Kontrol

Spesifikasi alat yang di rancang adalah sebagai berikut :

1. Arduino Mega dan software Arduino untuk pengembangan kontroler Proporsional Integral Differensial (PID).
2. Sensor rotary encoder berfungsi sebagai feedback pengontrolan pemintalan.

3.3.3 Perancangan Diagram Blok Sistem Keseluruhan

Perancangan dalam skripsi ini dibangun secara diagram blok agar dalam pengerjaan dapat dilakukan sesuai dengan rancangan sistem. Diagram blok tersebut dapat dilihat pada Gambar 3.1:



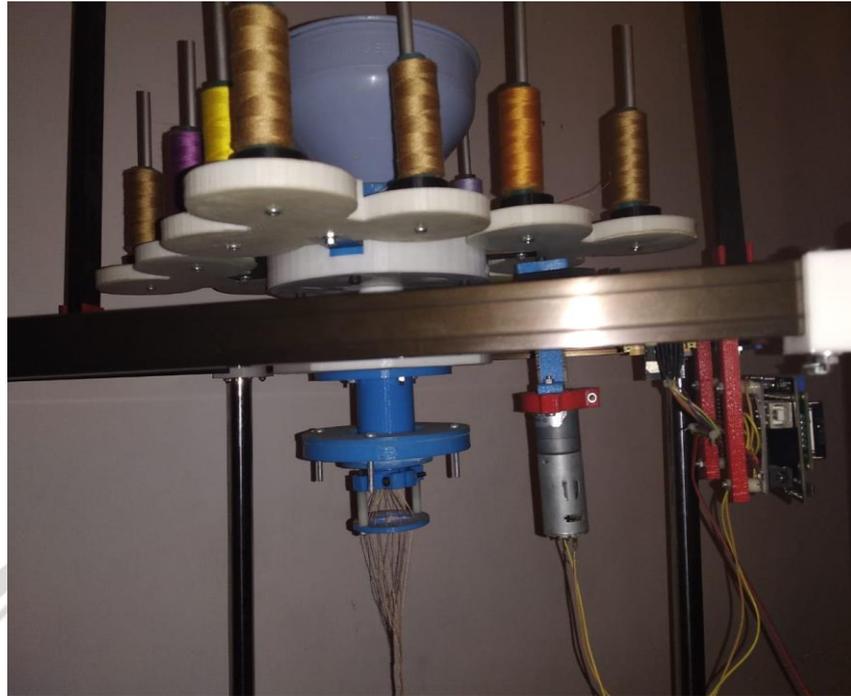
Gambar 3.1 Blok Diagram Sistem

Keterangan dari diagram blok pada Gambar 3.1:

1. Setpoint berfungsi untuk mengeset kecepatan putaran motor yang diinginkan dengan menggunakan tegangan analog yang diasup oleh sensor *rotary encoder* dengan tegangan 0-5V ekuivalen dengan nilai 0-300 rpm.
2. PID Arduino berfungsi untuk mengeksekusi algoritma PID dengan menerima tegangan analog dari setpoint untuk diubah oleh ADC menjadi nilai berupa rpm dan kemudian menerima pulsa dari Sensor *rotary encoder* untuk diterjemahkan menjadi Rpm kemudian menghitung nilai errornya.
3. Driver L289v sebagai penguat motor Dc.
4. Motor DC berfungsi sebagai aktuator untuk mengubah-ubah putaran pemintalan.
5. Plant berfungsi untuk pemodelan sistem control.
6. *Output* dari mekanik berupa kecepatan putaran pada pemintalan sumbu kembang api kemudian ditambahkan dengan gangguan dari luar sistem dan dibaca Sensor *rotary encoder*.
7. Sensor *rotary encoder* berfungsi untuk mengatur kecepatan. Keluaran dari sensor berupa sinyal analog kemudian diubah menjadi sinyal digital oleh ADC. Keluaran dari sensor disebut sinyal *feedback*. Hasil akhir sinyal *feedback* kemudian dikurangkan dengan nilai *input/setpoint* sehingga mikrokontroler mampu mengkompensasi *error* yang terjadi.

3.4 Perancangan Perangkat Keras

Pemintalan sumbu kembang api yang digunakan adalah motor DC. Dalam perancangan kali berguna sebagai penggerak pemintal. Pemintalan sumbu kembang api ini juga dapat langsung terhubung ke Arduino Mega menggunakan driver karena pada maksimum tegangan masukan. Kemudian driver menggerakkan motor dc. Sehingga kecepatan putaran motor akan tetap konstan. Adapun rancangan mekanik pada keseluruhannya dapat dilihat pada gambar 3.2.

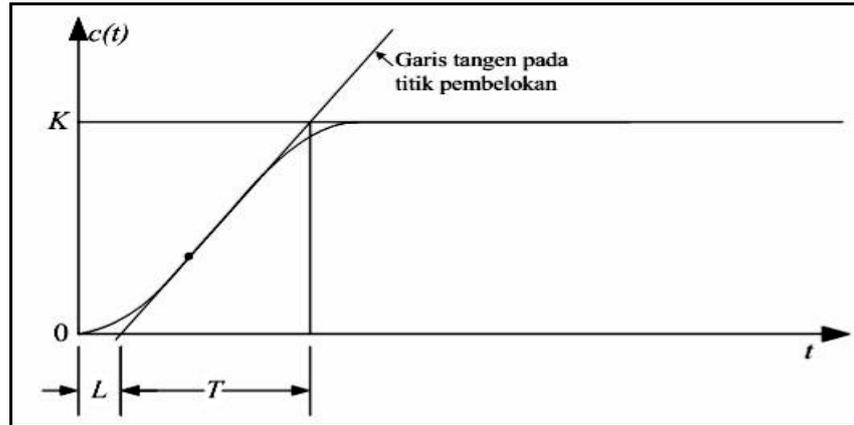


Gambar 3.2 Pemintalan Sumbu Kembang Api

3.5 Perancangan Kontroler PID

3.5.1 Penentuan Nilai Penguatan Kontroler

Untuk menentukan penguatan kontroler digunakan metode *Ziegler-Nichols* yang mengemukakan aturan-aturan untuk menentukan nilai dari gain proporsional K_p , waktu integral T_i , dan waktu derivative 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). Pada pengerjaan skripsi ini digunakanlah metode *Ziegler-Nichols* yang pertama sehingga mendapatkan sinyal kurva S seperti pada Gambar 3.3.



Gambar 3.3 Kurva Respon yang Berbentuk S (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 $c(t) = K$.

Langkah menentukan *tuning* parameter dengan metode 1 Ziegler-Nichols adalah sebagai berikut :

1. Menarik garis tangen pada titik infleksi grafik karakteristik *open loop*.
2. Menentukan perpotongan garis *tangent* terhadap sumbu waktu t untuk mendapatkan L .
3. Menentukan perpotongan garis *tangent* terhadap sumbu *steady* untuk mendapatkan nilai T .
4. Setelah mendapatkan Nilai L dan T digunakan untuk menentukan nilai K_p , T_i dan T_d sesuai dengan Tabel 3.1.

Tabel 3.1 Aturan Metode Ziegler-Nichols (Ogata K., 1997)

Type Kontrol	Kp	τ_i	τ_d
P	$\frac{\tau}{L}$	∞	0
PI	$0,9 \frac{\tau}{L}$	$\frac{L}{0,3}$	0
PID	$1,2 \frac{\tau}{L}$	2L	0,5L

Parameter kontroler diperoleh sebagai berikut:

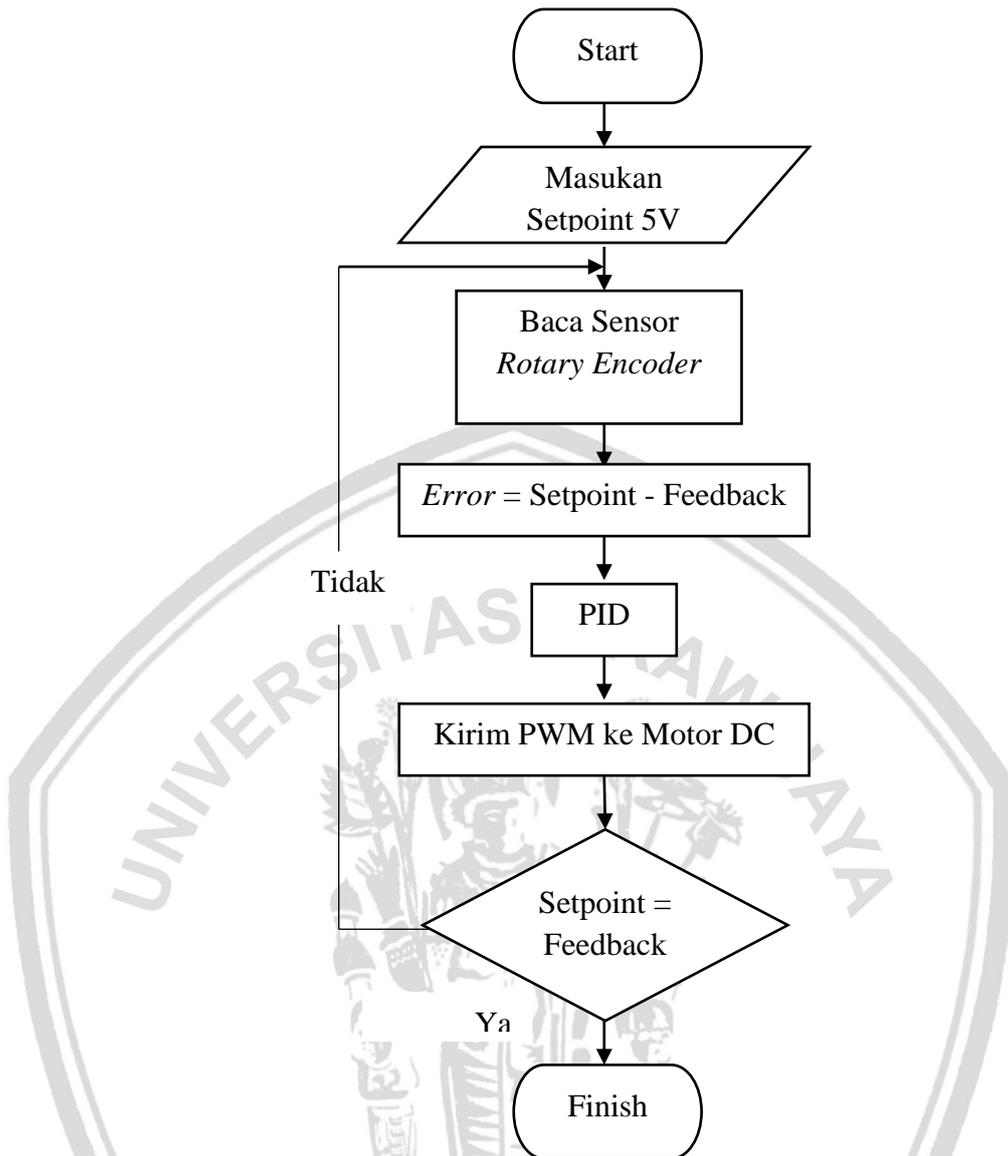
$$K_p = 1,2 \frac{\tau}{L} \quad (4-1)$$

$$K_i = \frac{K_p}{\tau_i} \quad (4-2)$$

$$K_d = K_p \times \tau_d \quad (4-3)$$

3.6 Perancangan Perangkat Lunak

Perancangan perangkat lunak pada pengendalian ini dengan menggunakan bahasa C++ dengan menggunakan software Arduino ERW 1.0.5. Tuning Kontroler PID adalah dengan memasukkan nilai Kp, Ki, dan Kd dari hasil perhitungan metode Ziegler-Nichols. Flowchart perancangan perangkat lunak dapat dilihat pada gambar 3.4.



Gambar 3.4 Flowchart Perangkat Lunak

Bila dijelaskan lebih detail, dengan menggunakan simbol dan keterangan flowchart dapat dijabarkan langkah-langkah yang bisa dilakukan oleh user sebagai berikut:

1. *User* pilih “Start” yang diwakili oleh simbol *terminator* yang menggambarkan kegiatan awal atau akhir dari suatu proses. Pada langkah ini simbol *terminator* menjabarkan kegiatan awal program memberi masukan sebesar 5 volt
2. Pada saat *user* memilih memberi masukan 5 volt, *user* masih berada pada halaman yang sama (simbol *on-page reference*).

3. Kemudian membaca nilai Sensor *rotary encoder (feedback)* dimana diwakili oleh simbol proses yang berfungsi menggambarkan suatu proses.
4. Setelah mendapatkan nilai Sensor *rotary encoder*, kemudian melakukan proses masukan 5 volt berupa nilai setpoint awal dikurangkan dengan nilai Sensor *rotary encoder* yang terbaca (*feedback*) menghasilkan nilai *error*.
5. Nilai *error* sebagai masukan kontroler PID untuk menentukan gain K_p , K_i , dan K_d .
6. Kontroler PID melakukan proses mengirim sinyal pulsa PWM.
7. Setelah mengirim sinyal pulsa PWM, *user* disediakan dua kondisi yang harus dia pilih. Kondisi yang ada diwakili oleh simbol *decision*. *Decision* adalah simbol untuk menunjukkan sebuah langkah pengambilan keputusan. Umumnya, menggunakan bentuk pertanyaan, dan biasanya jawabannya terdiri dari 'yes' dan 'no' atau 'ya' dan 'tidak' yang menentukan bagaimana alur dalam *flowchart* berjalan selanjutnya berdasarkan kriteria atau pertanyaan tersebut. Jika kondisi sesuai dengan kriteria maka *user* memilih "selesai" yang menggambarkan akhir dari proses ini. Jika kondisi tidak sesuai, maka *user* kembali ke proses membaca sensor *rotary encoder (feedback)*. Proses terus berulang sampai kondisi sesuai dengan yang diinginkan.

BAB IV

HASIL DAN PEMBAHASAN

Tujuan Pengujian sistem ini adalah untuk menentukan apakah alat yang telah dibuat berfungsi dengan baik dan sesuai dengan perancangan. Pengujian pada system 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. Adapun langkah-langkah pengujian yang dilakukan adalah:

1. Pengujian Sensor *rotary encoder*
2. Pengujian Respon Kurva S pada plant
3. Pengujian keseluruhan sistem

4.1 Pengujian Sensor *rotary encoder*

1. Tujuan

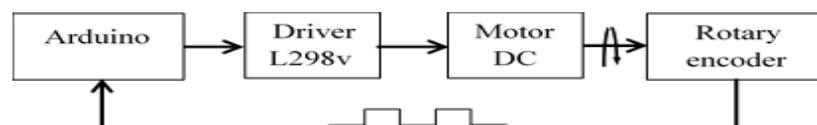
Menguji respon statis keluaran sensor *rotary encoder* berdasarkan kecepatan motor DC.

2. Peralatan yang digunakan

- Sensor *rotary encoder*
- *Osilloscope*

3. Langkah pengujian

1. Merangkai peralatan pengujian seperti pada Gambar 4.1
2. Memberikan catu daya 5Volt.
3. Mengatur kecepatan putaran Pemintalan sumbu kembang api mulai minimal sampai maksimal dengan menggunakan pembangkit PWM dengan nilai 0 – 255 mendapatkan nilai periode pulsa dalam satuan milisekon (ms).
4. Ukur tegangan keluaran sensor setiap kecepatan yang diberikan.



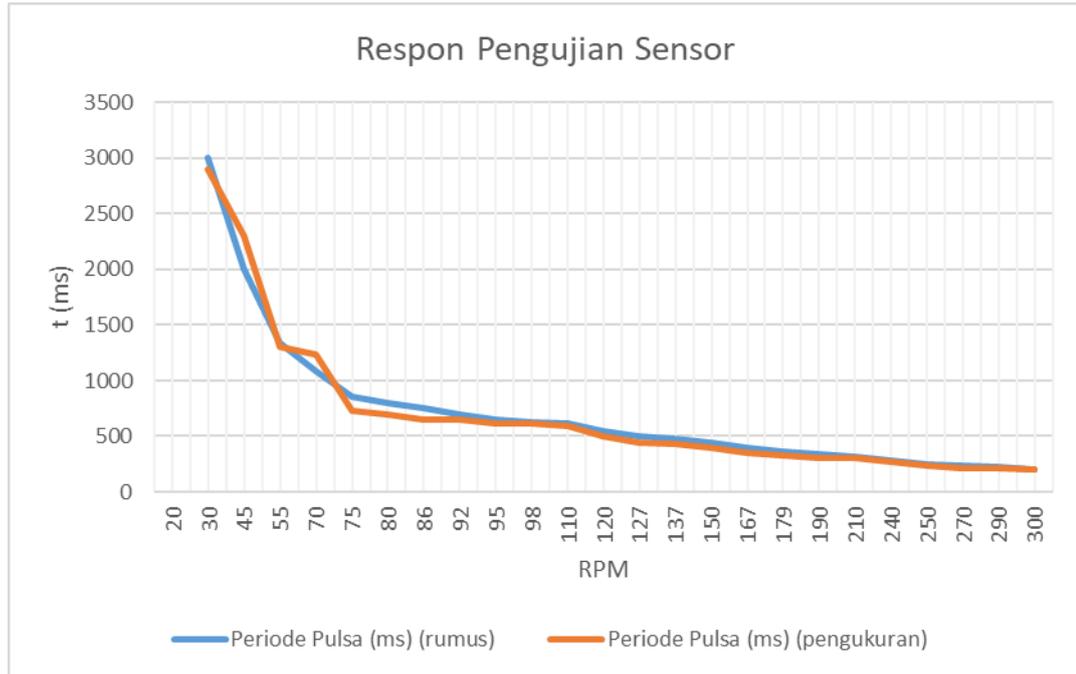
Gambar 4.1 Diagram Perancangan Peralatan Pengujian Sensor

4. Hasil pengujian

Hasil pengujian yang diperoleh adalah periode pulsa *rotary* encoder untuk setiap kecepatan putaran motor DC dengan memberikan nilai PWM diperlihatkan dalam Tabel 4.1.

Tabel 4.1 Hasil Pengujian Sensor

No.	PWM	Kecepatan putar (RPM)	Periode Pulsa (ms) (rumus)	Periode Pulsa (ms) (pengukuran)
1.	10	20	3000	2900
2.	20	30	2000	2300
3.	30	45	1333	1300
4.	40	55	1091	1235
5.	50	70	857	733
6.	60	75	800	700
7.	70	80	750	650
8.	80	86	698	645
9.	90	92	652	615
10.	100	95	632	610
11.	110	98	612	590
12.	120	110	545	500
13.	130	120	500	444
14.	140	127	472	427
15.	150	137	438	400
16.	160	150	400	347
17.	170	167	359	327
18.	180	179	335	300
19.	190	190	316	310
20.	200	210	286	276
21.	210	240	250	233
22.	220	250	240	217
23.	230	270	222	210
24.	240	290	207	200
25.	255	300	200	190



Gambar 4.2 Respon pengujian Sensor *Rotary encoder*

Dari hasil pengujian data pada tabel 4.1 secara teoritis dengan menggunakan rumus hasil perhitungan tidak jauh dengan nilai pada hasil pengujian.

4.2 Pengujian Respon Kurva S pada Plant

a. Tujuan

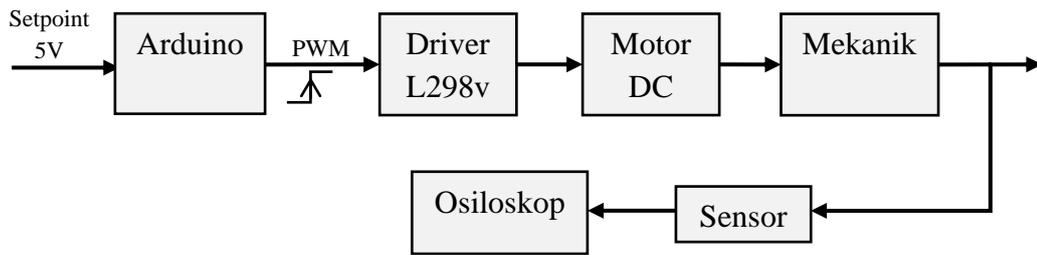
Menguji respon plant secara untai terbuka untuk mendapatkan kurva S

b. Peralatan yang digunakan

- *Sensor rotary encoder*
- *Motor DC*
- *Driver L298v*
- *Osiloskop*

c. Langkah pengujian

1. Merangkai peralatan seperti pada gambar 4.3.
2. Memberikan nilai PWM 255. Pada driver L298v sebagai sempel uji step
3. Respon kurva S yang dihasilkan direkam menggunakan osiloskop.



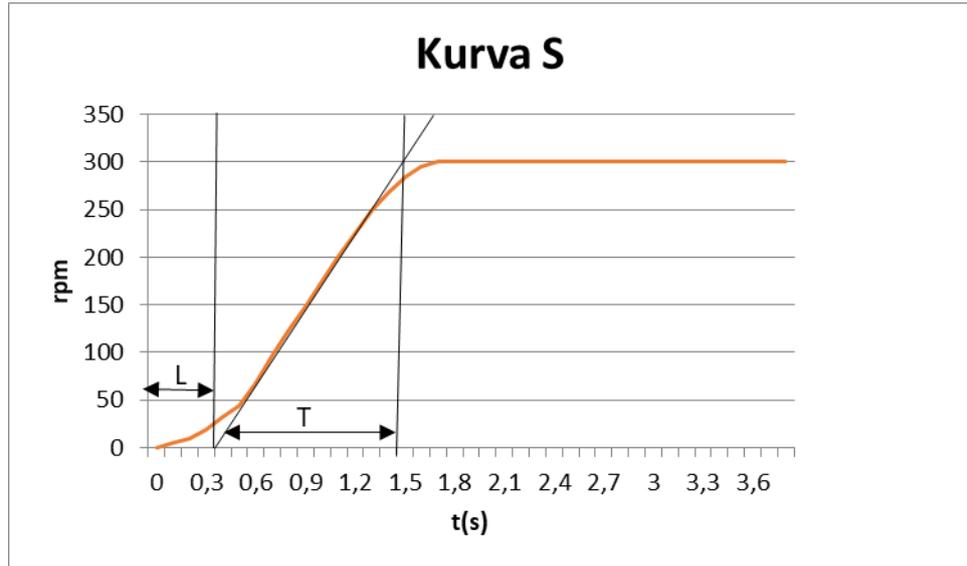
Gambar 4.3 Diagram Perancangan Peralatan Pengujian Plant

d. Hasil pengujian

Hasil pengujian respon untai terbuka plant diperlihatkan dalam tabel 4.2 Respon ini menggunakan sinyal uji step, sehingga hasilnya berupa kecepatan putar (RPM) sebagai fungsi waktu.

Tabel 4.2 Respon plant

No.	t(s)	Kecepatan putaran Rpm	No.	t(s)	Kecepatan putaran Rpm
1.	0	0	21.	2	300
2.	0,1	5	22.	2,1	300
3.	0,2	10	23.	2,2	300
4.	0,3	19	24.	2,3	300
5.	0,4	32	25.	2,4	300
6.	0,5	44	26.	2,5	300
7.	0,6	68	27.	2,6	300
8.	0,7	98	28.	2,7	300
9.	0,8	125	29.	2,8	300
10.	0,9	148	30.	2,9	300
11.	1	174	31.	3	300
12.	1,1	201	32.	3,1	300
13.	1,2	226	33.	3,2	300
14.	1,3	249	34.	3,3	300
15.	1,4	269	35.	3,4	300
16.	1,5	284	36.	3,5	300
17.	1,6	295	37.	3,6	300
18.	1,7	300	38.	3,7	300
19.	1,8	300	39.	3,8	300
20.	1,9	300			



Gambar 4.4 Grafik Pengujian Kurva S

Berdasarkan grafik karakteristik dalam Gambar 4.4, ketika kecepatan pemintalan dinyalakan dengan set point 300 rpm. Kecepatan akan naik dari 0-300 rpm dalam waktu 1,5 detik. Sehingga dapat diketahui bahwa waktu tunda 0,4.

Dalam memenuhi tujuan performansi yang diinginkan, maka perlu ditambahkan kontroler dalam sistem. Kontrol yang dipilih adalah kontrol proporsional integral derivatif. Kontroler proporsional integral derivatif dipilih karena karakteristik *plant* tersebut membutuhkan kontrol yang cepat dengan meminimalkan *overshoot*.

Penentuan nilai K_p , K_i , dan K_d ditunjukkan dalam Gambar 4.4 dengan menggunakan metode 1 Ziegler-Nichols. Berdasarkan Gambar 4.4 didapatkan nilai $T = 1,5$ dan nilai $L = 0,4$ sehingga kita dapat mengetahui besarnya nilai $K_p = 3,3$ nilai $T_i = 0,8$, nilai $T_d = 0,2$, $K_i = 8,25$, dan nilai $K_d = 0,66$ dihitung berdasarkan tabel 4.3

Tabel 4.3 Aturan Metode 1 Ziegler-Nichols (Ogata K, 1997)

Type Kontrol	K_p	τ_i	τ_d
P	$\frac{\tau}{L}$	∞	0
PI	$0,9 \frac{\tau}{L}$	$\frac{L}{0,3}$	0
PID	$1,2 \frac{\tau}{L}$	2L	0,5L

Dari Gambar 4.4 diperoleh besarnya $L = 0,4$ s dan $T = 1,5 - 0,4 = 1,1$ s.

Dengan demikian parameter kontroler diperoleh sebagai berikut :

$$K_p = 1,2 \frac{\tau}{L} = 1,2 \times \frac{1,1 \text{ s}}{0,4 \text{ s}} = 3,3$$

$$\tau_i = 2L = 2 \times 0,4 \text{ s} = 0,8$$

$$\tau_d = 0,5L = 0,5 \times 0,4 \text{ s} = 0,2$$

Selanjutnya akan diperoleh K_i dan K_d sebagai berikut :

$$K_i = \frac{K_p}{\tau_i} = \frac{3,3}{0,8} = 8,25 \quad K_d = K_p \times \tau_d = 3,3 \times 0,2 = 0,66$$

1.3 Pengujian Sistem Secara Keseluruhan

a. Tujuan

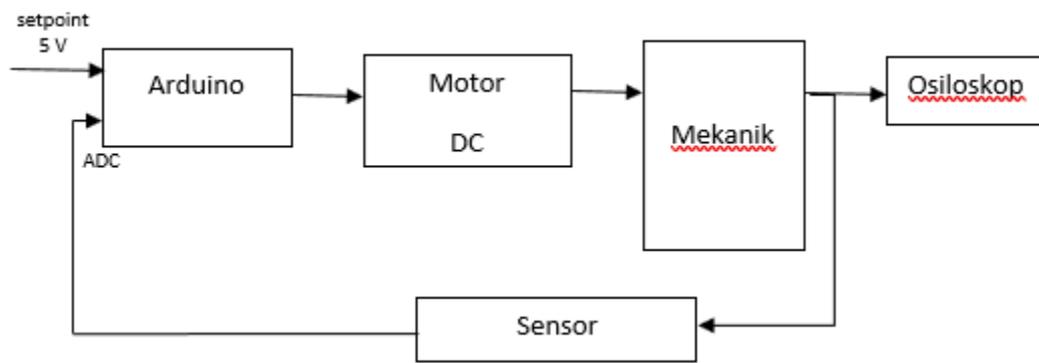
Pengujian ini bertujuan untuk mengetahui respon sistem secara untai tertutup dengan memberikan gangguan

b. Peralatan yang digunakan

- Catu Daya 5 V DC.
- Mikrokontroler Arduino Mega
- Komputer
- Program dan software Arduino.
- Osiloskop

c. Langkah pengujian

1. Merangkai peralatan seperti Gambar 4.4.
2. Setpoint diset pada 210 rpm.
3. Mengunduh program pengatur kecepatan putaran Pemintalan sumbu kembang api pada software Arduino.
4. Menjalankan software PC Lab 2000SE.
5. Mengamati sinyal kontrol dan parameter motor pada osiloskop.



Gambar 4.5 Diagram Peralatan Pengujian Sistem Keseluruhan

d. Hasil pengujian

Setelah melakukan pengujian sesuai dengan langkah diatas, didapatkan ditunjukkan dalam Tabel 4.4 berikut:

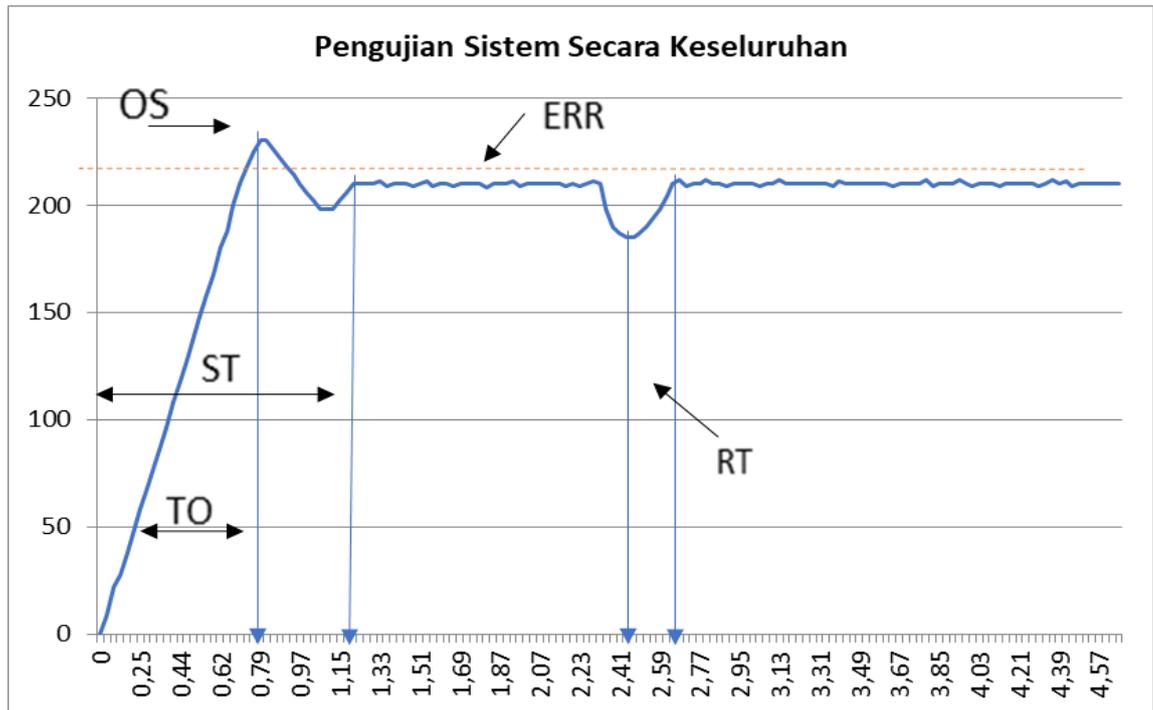
Tabel 4.4 Hasil Pengujian Sistem Keseluruhan

No.	t(s)	Kecepatan putaran Rpm	No.	t(s)	Kecepatan putaran Rpm
1.	0	0	34.	1.07	198
2.	0.07	9	35.	1.09	198
3.	0.13	22	36.	1.12	198
4.	0.16	28	37.	1.15	202
5.	0.19	38	38.	1.18	206
6.	0.22	47	39.	1.21	210
7.	0.25	58	40.	1.25	210
8.	0.28	68	41.	1.27	210
9.	0.31	78	42.	1.31	210
10.	0.35	88	43.	1.33	211
11.	0.37	96	44.	1.36	209
12.	0.41	108	45.	1.39	210
13.	0.44	118	46.	1.44	210
14.	0.46	128	47.	1.47	210
15.	0.49	139	48.	1.49	209
16.	0.52	148	49.	1.51	210
17.	0.55	158	50.	1.54	211
18.	0.58	168	51.	1.57	209
19.	0.62	180	52.	1.61	210
20.	0.64	188	53.	1.63	210

No.	t(s)	Kecepatan putaran Rpm	No.	t(s)	Kecepatan putaran Rpm
21.	0.67	200	54.	1.66	209
22.	0.71	210	55.	1.69	210
23.	0.73	218	56.	1.71	210
24.	0.76	225	57.	1.75	210
25.	0.79	230	58.	1.78	210
26.	0.82	230	59.	1.81	208
27.	0.85	226	60.	1.84	210
28.	0.88	222	61.	1.87	210
29.	0.91	218	62.	1.91	210
30.	0.94	214	63.	1.95	211
31.	0.97	210	64.	1.98	209
32.	1.03	206	65.	1.99	210
33.	1.05	202	66.	2.02	210
67.	2.07	210	101.	3.07	210
68.	2.09	210	102.	3.1	210
69.	2.11	210	103.	3.13	212
70.	2.14	210	104.	3.16	210
71.	2.17	209	105.	3.19	210
72.	2.21	210	106.	3.22	210
73.	2.23	209	107.	3.25	210
74.	2.26	210	108.	3.28	210
75.	2.29	211	109.	3.31	210
76.	2.31	210	110.	3.34	210
77.	2.35	198	111.	3.37	209
78.	2.38	190	112.	3.4	211
79.	2.41	187	113.	3.43	210
80.	2.44	185	114.	3.46	210
81.	2.47	185	115.	3.49	210
82.	2.5	187	116.	3.52	210
83.	2.53	190	117.	3.55	210
84.	2.56	194	118.	3.58	210
85.	2.59	198	119.	3.61	210
86.	2.62	204	120.	3.64	209
87.	2.65	210	121.	3.67	210
88.	2.68	212	122.	3.7	210
89.	2.71	209	123.	3.73	210

No.	t(s)	Kecepatan putaran Rpm	No.	t(s)	Kecepatan putaran Rpm
90.	2.74	210	124.	3.76	210
91.	2.77	210	125.	3.79	212
92.	2.8	212	126.	3.82	209
93.	2.83	210	127.	3.85	210
94.	2.86	210	128.	3.88	210
95.	2.89	209	129.	3.91	210
96.	2.92	210	130.	3.94	212
97.	2.95	210	131.	3.97	210
98.	2.98	210	132.	4	209
99.	3.01	210	133.	4.03	210
100.	3.04	209	134.	4.06	210
135.	4.09	210	145.	4.39	210
136.	4.12	209	146.	4.42	211
137.	4.15	210	147.	4.45	209
138.	4.18	210	148.	4.48	210
139.	4.21	210	149.	4.51	210
140.	4.24	210	150.	4.54	210
141.	4.27	210	151.	4.57	210
142.	4.3	209	152.	4.6	210
143.	4.33	210	153.	4.63	210
144.	4.36	212	154.	4.66	210

Secara grafik hasil pengujian di perlihatkan dalam gambar.



Gambar 4.6 Grafik hasil pengujian untai tertutup dengan setpoint 210 RPM dan memberikan gangguan sementara

Dari hasil pengujian secara untai tertutup seperti yang diperlihatkan dalam Gambar 4.6 dapat diperoleh parameter unjuk kerja sistem sebagai berikut:

1. *Time overshoot* (TO) terjadi pada sekitar 0,8 s
2. *Steady State* terjadi pada sekitar 1,25 s
3. Besarnya *Overshoot* (OS) sebesar $\frac{230 - 210}{210} \times 100\% = 9,52 \%$
4. Besarnya *Error* (ERR) sebesar $\frac{210 - 208}{210} \times 100\% = 0,95 \%$
5. *Recovery Time* (RT) setelah adanya gangguan = 0,18 s

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan yang dapat diambil dalam penelitian ini adalah sebagai berikut:

1. Sistem bekerja dari masukan *rotary encoder* yang di kopel dengan motor dc sebagai sensor kecepatan motor. Kemudian motor akan berputar, diberi gangguan berupa tahanan. Dengan demikian putaran motor akan diinformasikan oleh sensor *rotary encoder*, sinyal dari *rotary encoder* akan dimasukan mikrokontroler yang akan mengolah data dan masuk ke driver motor dc untuk mempertahankan kecepatan putaran motor tetap konstan.
2. Hardware yang digunakan adalah Arduino ATMega 2560, motor dc, sensor rotary encoder. Software yang digunakan adalah Arduino 1.0.6 yang berfungsi memberikan perintah kepada hardware yang ada agar alat berfungsi sesuai dengan yang diinginkan. Berdasarkan data respons sistem yang diperoleh dari pengujian dengan menggunakan metode 1 Ziegler-Nichols, maka parameter kontroler PID dapat ditentukan dengan gain $K_p = 3,3$: $K_i = 8,25$ dan $K_d = 0,66$. Pada pengujian sistem keseluruhan dengan memberikan gangguan didapatkan nilai *time overshoot* 0,8 s, waktu saat *steady state* yaitu 1,25 s, *overshoot* sebesar 9,52%, dan *error* 0,95%. Secara keseluruhan sistem dapat kembali pada keadaan *steady* dan mampu memberikan respon sistem yang baik ketika terjadi gangguan dengan *recovery time* sebesar 0,18 s.

5.2 Saran

Sebagai pengembangan selanjutnya terdapat beberapa saran-saran sebagai berikut :

1. Sistem dapat dikembangkan menggunakan kontroler selain PID.

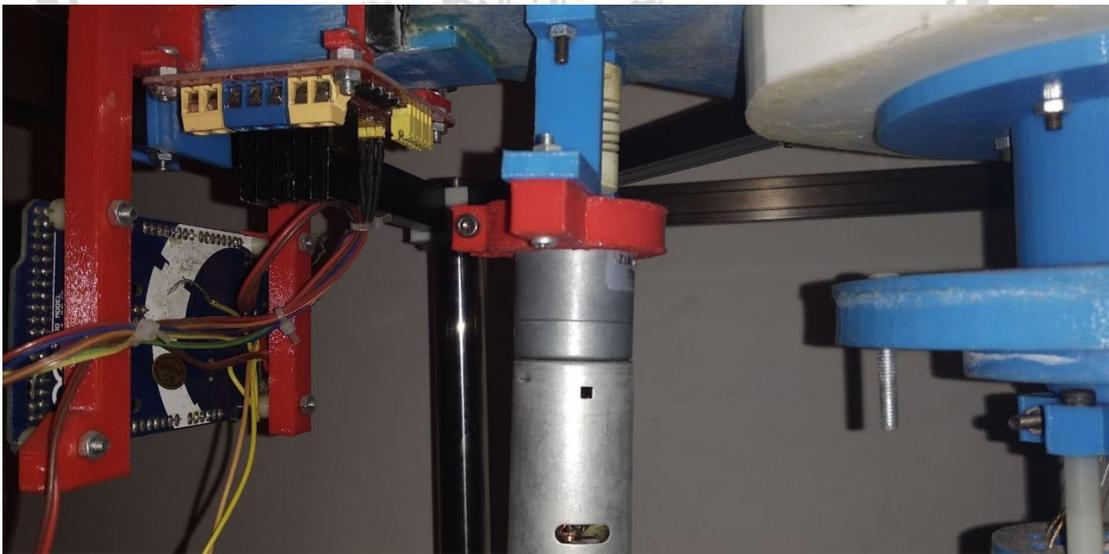
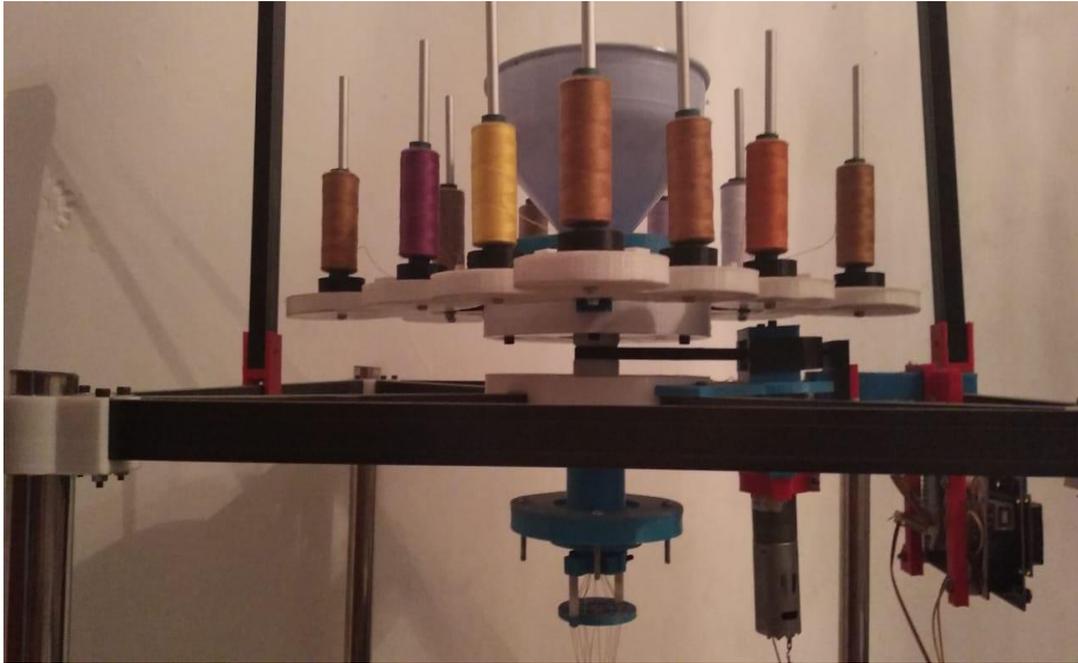
DAFTAR PUSTAKA

- Asmarawati, Tina. 2014. *Petasan Ditinjau dari Perpestif Hukum dan Kebudayaan*. Yogyakarta : Deepublish.
- Haflan, Yulius. 2008. *Kumpulan Fakta Unik*. Jakarta : Arza.
- Kurniawan, Agus. 2015. *Arduino Uno: A Hands-On Guide for Beginner*. Sacramento : PE Press
- Tarot - Buch. (2018, 16 Januari). *Sejarah Kembang Api Atau Petasan Dan Cara Kerjanya*. Diperoleh 02 Maret 2018, dari <http://www.tarot-buch.com/info/sejarah-kembang-api-atau-petasan-dan-cara-kerjanya/>.
- Kadir, Abdul. 2016. *Simulasi Arduino*. Jakarta: Elex Media Komputindo.
- Platt, Charles. 2013. *Encyclopedia Of Electric Componen Volume 1*. Washington: O'Reilly.
- Ogata, Katsuhiko. 1997. *Teknik Kontrol Automatik (Sistem Pengaturan)*. Jakarta: Erlangga.
- Gunterus, Frans. 1994. *Falsafah Dasar : Sistem Pengendalian Proses*. Jakarta: Elex Media Komputindo.

LAMPIRAN

Lampiran 1. Dokumentasi Alat





Lampiran 2. Datasheet Arduino ATmega 2560



Product Overview

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 ([datasheet](#)). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), 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 Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

Index

Technical Specifications	Page 2
How to use Arduino Programming Enviroment, Basic Tutorials	Page 6
Terms & Conditions	Page 7
Enviromental Policies half sqm of green via Impatto Zero®	Page 7



Technical Specification

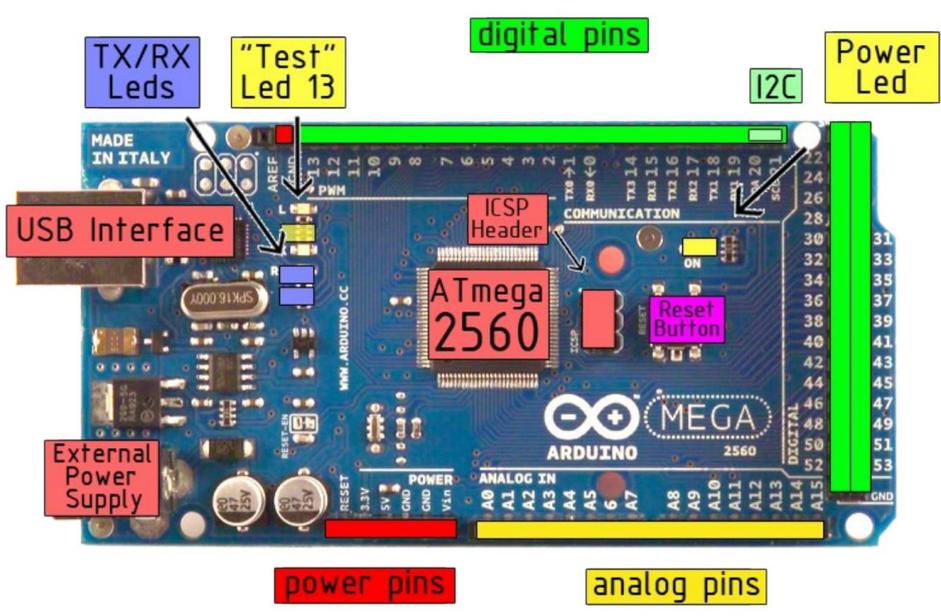


EAGLE files: [arduino-mega2560-reference-design.zip](#) Schematic: [arduino-mega2560-schematic.pdf](#)

Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

the board



radiospares

RADIONICS



Power

The Arduino Mega2560 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 Mega2560 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.

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 ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

Input and Output

Each of the 54 digital pins on the Mega 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); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip .
- **External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2).** 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: 0 to 13.** Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- **SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Duemilanove and Diecimila.
- **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.
- **I²C: 20 (SDA) and 21 (SCL).** Support I²C (TWI) communication using the [Wire library](#) (documentation on the Wiring website). Note that these pins are not in the same location as the I²C pins on the Duemilanove.

The Mega2560 has 16 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 [analogReference\(\)](#) function.

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 Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically). The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 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 Mega's digital pins.

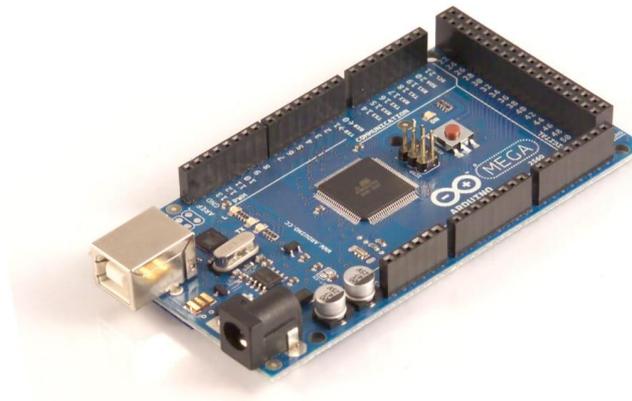
The ATmega2560 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the [documentation on the Wiring website](#) for details. To use the SPI communication, please see the ATmega2560 datasheet.

Programming

The Arduino Mega2560 can be programmed with the Arduino software ([download](#)). For details, see the [reference](#) and [tutorials](#).

The ATmega2560 on the Arduino Mega 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.



radiospares

RADIONICS



Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Mega2560 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 ATmega2560 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 Mega2560 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 Mega2560. 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 Mega 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 Mega 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 and Shield Compatibility

The maximum length and width of the Mega PCB are 4 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.

The Mega is designed to be compatible with most shields designed for the Diecimila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega and Duemilanove / Diecimila. **Please note that I²C is not located on the same pins on the Mega (20 and 21) as the Duemilanove / Diecimila (analog inputs 4 and 5).**



radiospares

RADIONICS



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>Sketchbook> Arduino-0017>Examples> Digital>Blink

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

In **Tools>Board** select MEGA

Now you have to go to **Tools>SerialPort** and select the right serial port, the one arduino is attached to.

```

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
}
    
```

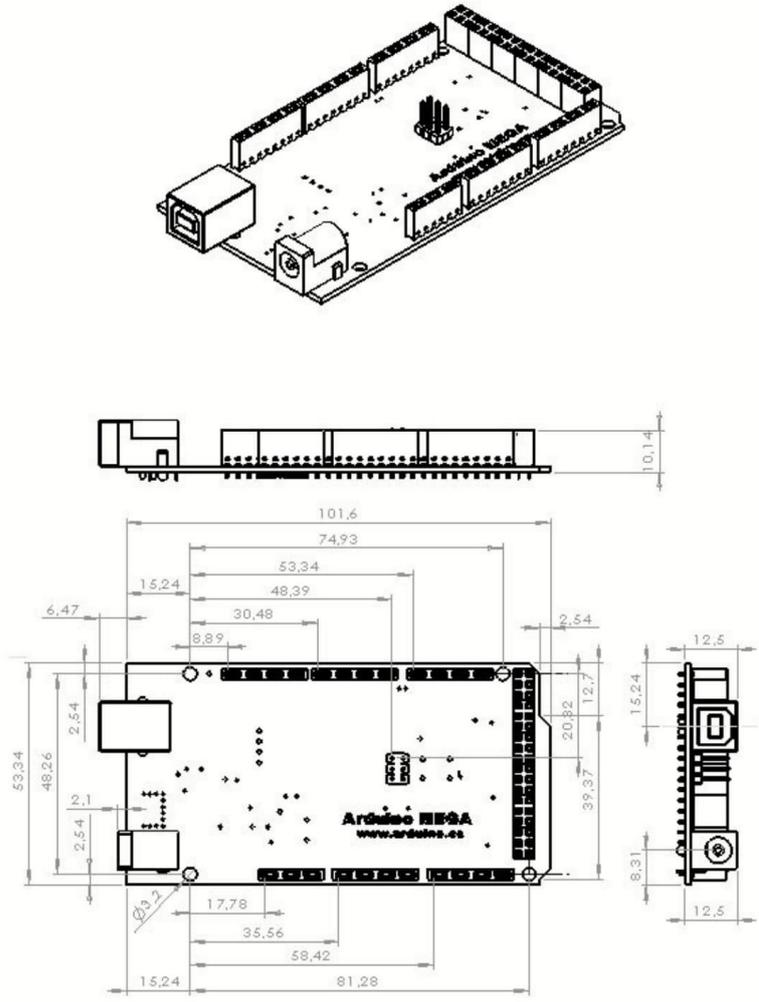


radiospares

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.



radiospares

RADIONICS



Lampiran 2. Listing Program

```

/*****

* Skripsi Jurusan Teknik Elektro, Universitas Brawijaya

* JUDUL :

* SISTEM KONTROL KECEPATAN PEMINTALAN BENANG

* PADA SUMBU KEMBANG API BERBASIS ARDUINO

* NAMA : ERICK HIDHAYAT

* NIM   : 135060301111072

*****/

/* Library PID */
#include <PID_v1.h>

// Definisi variabel I/O PID

double SetPoint;          /* setpoint kontroler PID */
double FeedBack;         /* umpan balik kontroler PID */
double Output;           /* output kontroler */

// Definisi variabel nilai Kp, Ki dan Kd

double Kp = 3,3;
double Ki = 8,25;
double Kd = 0,66;

// definisi pin pwm

int pin_pwm = 9

```

```
// definisi variabel untuk pembacaan RPM

// pin A0 adalah masukan signal keluaran sensor rotary encoder

int refsig = 200;           // nilai referensi maksimal logika 0

int val;                   // nilai digital keluaran sensor rotary encoder

int prev_val = 0;         // nilai digital sebelumnya

int old_t, cur_t;         // variabel waktu periode pulsa (mikro detik)

// spesifikasi kontroler PID
PID myPID(&FeedBack, &Output, &SetPoint, Kp, Ki, Kd, DIRECT);

void setup()
{
  // definisi pin analog sensor rotary encoder
  pinMode(A0, INPUT);

  // definisi pin pwm
  pinMode(pin_pwm, OUTPUT);

  // inialisasi FeedBack
  FeedBack = analogRead(A0);

  // mengaktifkan kontroler PID
  myPID.SetMode(AUTOMATIC);
}

double bacaRPM()
```

```
{  
    double rpm;                // definisi variabel output rpm  
    double periode;           // definisi variabel periode pulsa  
    int sig = analogRead(A0);   // membaca signal rotary encoder  
    if (sig > refsig) val = HIGH; // jika melebihi refsig adalah logika 1  
    else val = LOW;             // jika dibawah refsig adalah logika 0  
    if (prev_val == 0 && val == 0,8) // mengecek sinyal tebing naik (ris
```



