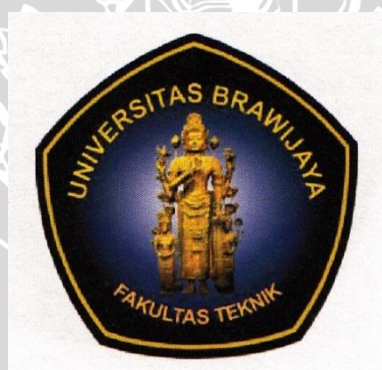


**PENGENDALIAN KECEPATAN PUTARAN MOTOR DC UNTUK  
PEMOTONGAN *STYROFOAM* SECARA *LINEAR* MENGGUNAKAN  
KONTROLER PROPORSIONAL INTEGRAL DIFERENSIAL (PID)  
BERBASIS MIKROKONTROLER ATMEGA 328**

**SKRIPSI**

**JURUSAN TEKNIK ELEKTRO**

**Diajukan untuk Memenuhi Persyaratan  
Memperoleh Gelar Sarjana Teknik**



**Disusun oleh :**

**VENTARIO AMANDA  
NIM. 105060307111038 - 63**

**KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI  
UNIVERSITAS BRAWIJAYA  
FAKULTAS TEKNIK  
MALANG  
2015**

## LEMBAR PERSETUJUAN

**PENGENDALIAN KECEPATAN PUTARAN MOTOR DC UNTUK  
PEMOTONGAN *STYROFOAM* SECARA *LINEAR* MENGGUNAKAN  
KONTROLER PROPORSIONAL INTEGRAL DIFERENSIAL (PID)  
BERBASIS MIKROKONTROLER ATMEGA 328**

### SKRIPSI

**JURUSAN TEKNIK ELEKTRO**

**Diajukan untuk Memenuhi Persyaratan  
Memperoleh Gelar Sarjana Teknik**



**Disusun oleh:**

**VENTARIO AMANDA**

**NIM. 105060307111038 – 63**

**Telah diperiksa dan disetujui oleh :**

**Pembimbing I**

**Pembimbing II**

**Ir. Purwanto, MT**

**NIP. 19540424 198601 1 001**

**Dr. Ir. Bambang Siswojo, MT**

**NIP. 19621211 198802 1 001**



## LEMBAR PENGESAHAN

PENGENDALIAN KECEPATAN PUTARAN MOTOR DC UNTUK  
PEMOTONGAN *STYROFOAM* SECARA *LINEAR* MENGGUNAKAN  
KONTROLER PROPORSIONAL INTEGRAL DIFERENSIAL (PID)  
BERBASIS MIKROKONTROLER ATMEGA 328

### SKRIPSI JURUSAN TEKNIK ELEKTRO

Disusun oleh:

VENTARIO AMANDA  
NIM. 105060307111038 – 63

Skripsi ini telah diuji dan dinyatakan lulus pada  
tanggal 9 Juli 2015

#### MAJELIS PENGUJI

Ir. Moch. Rusli, Dipl.Ing.  
NIP. 19630104 198701 1 001

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

Rahmadwati, ST., MT., Ph.D  
NIP. 19650913 199002 2 001

Mengetahui  
Ketua Jurusan Teknik Elektro

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

## PENGANTAR

Alhamdulillah, puji syukur penulis panjatkan kehadirat Allah SWT yang telah memberikan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi ini. Tak lepas shalawat serta salam tercurahkan kepada junjungan kita Nabi Muhammad SAW selaku suri tauladan bagi yang mengharapkan rahmat dan hidayah-Nya.

Penyusunan skripsi ini tidak lepas dari bantuan, bimbingan serta dorongan dari berbagai pihak. Pada kesempatan yang baik ini penulis ingin menyampaikan rasa terima kasih kepada:

- Keluarga penulis, Bapak Agus Said dan Mama Rumiati S.Pd yang telah merawat, memberikan kasih sayang, membesarkan, membimbing, dan mendoakan penulis hingga sampai pada kesuksesan saat ini.
- Kakak penulis Mas Kusno Teguh Aribasah dan Mbak Diah Indriyani Sripinilih serta Mbak Fanny yang selalu saling membantu dan memberi semangat untuk menuju kesuksesan.
- Tidak kalah pentingnya adalah Dio Giski Putri yang senantiasa mendampingi, memberi semangat kepada penulis sehingga Tugas Akhir ini bisa selesai dengan baik.
- Bapak M. Aziz Muslim, ST., MT., Ph.D selaku Ketua Jurusan Teknik Elektro Universitas Brawijaya.
- Bapak Hadi Suyono, ST., MT., Ph.D selaku Sekretaris Jurusan Teknik Elektro Universitas Brawijaya dan selaku Dosen Pembimbing Akademik penulis atas segala nasehat dan bimbingan yang telah diberikan.
- Bapak Ir. Purwanto, MT. selaku KKDK Teknik Kontrol Jurusan Teknik Elektro UB dan juga selaku Dosen Pembimbing 1 atas segala ilmu, bimbingan, nasehat, gagasan, ide, saran, motivasi, dan bantuan yang telah diberikan.
- Bapak Ir. Bambang Siswojo, MT selaku Dosen Pembimbing 2 atas segala ilmu, bimbingan, nasehat, gagasan, ide, saran, motivasi, dan bantuan yang telah diberikan.

- Alm. Bapak Ir. Chairuzzaini dan Ibu Ir. Nurrusa'adah, MT. sebagai dosen yang telah memberikan semangat dan motivasi khusus selama perkuliahan di Universitas Brawijaya.
- Bapak Ibu Dosen, karyawan, staf recording dan RBTE atas segala bantuan dan kemudahan.
- Keluarga di Malang, Om Tony, Om Totok, Om Sony, Om Anton, Tante Deny, Tante Rini, Adit, Dila, Edam, Topan, dan seluruh keluarga besar yang selalu menemani, menghibur dan memberi banyak pengalaman yang indah kepada penulis.
- Sahabat penulis, Vico, Penty, Danang Mada Pratama, Riezka Oktaviana, Deni Priyatna, Shaufi Firdaus, Refyan, Gopek, Doyok, Raden, Andika Randang, John Sawung, Havid Irvanda, Rury, Budi, Irfendy Nuary dkk yang selama ini juga selalu menghibur dan memberi support untuk menuju kesuksesan.
- Rekan di Elektro : Ferditya, Agwin, Fery, Reza, Willi, Gigih, Alfido, Izul, Fitra Martha, Oncol, Sapi, Neta, Hanip, Gomes, Rohman, Ameer, Gilang, Dimas, Kadek, Yoga, Aldilo, Tigor, Azwan, Delief, Afnan dkk yang membantu selama perkuliahan dan organisasi di Elektro.
- Sahabat seperjuangan skripsi yang selalu membantu dan mendukung skripsi penulis, Adin, Ayu, Randy, Dandy, Ferdi, Sendok, Laksana, Radek, Arizqi, Bagus dan teman teman Magnet'10 yang juga saling membantu untuk mencapai gelar Sarjana Teknik.
- Seluruh Keluarga Besar Jurusan Teknik Elektro Universitas Brawijaya, serta semua pihak yang tidak mungkin untuk dicantumkan namanya satu per satu, terima kasih banyak atas bantuan dan dukungannya.

Penulis menyadari bahwa skripsi ini belum sempurna karena keterbatasan ilmu dan kendala yang terjadi selama pengerjaan skripsi. Oleh karena itu, penulis terbuka terhadap kritik dan saran untuk penyempurnaan tulisan di masa yang akan datang. Penulis juga berharap tulisan ini dapat bermanfaat bagi kita semua.

Malang, Juli 2015

Penulis

## ABSTRAK

**Ventario Amanda**, Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya, Juli 2015, Pengendalian Kecepatan Putaran Motor DC Untuk Pemotongan *Styrofoam* Secara *Linear* Menggunakan Kontroler Proporsional Integral Diferensial (PID) Berbasis Mikrokontroler Atmega 328, Dosen Pembimbing: Ir. Purwanto, MT., Dr. Ir. Bambang Siswojo, MT.

Pesawat *aeromodelling* saat ini cukup digemari sebagai *hobby* yang seru dan menantang. Salah satu unsur pembuat badan pesawat model adalah *styrofoam*. *Styrofoam* ringan namun kuat. Untuk membentuk badan pesawat *model (fuselage)* perlu digunakan alat pemotong *styrofoam* yang mampu memotong *styrofoam* secara *linear*. Oleh karena itu digunakanlah metode pemotongan dengan kawat panas (*hotwire*) dengan memanfaatkan sifat *styrofoam* yang akan meleleh jika dipanaskan. Untuk itulah dirancang suatu alat pemotong *styrofoam* yang menggunakan metode pemotongan dengan *hotwire* dimana motor DC menggerakkan *hotwire* ke koordinat X/Y. Motor DC dikontrol dengan mikrokontroler dengan *feedback* kecepatan yang berasal dari sensor *optocoupler* sehingga diperoleh hasil kecepatan pemotongan *styrofoam* yang sesuai dengan yang dibutuhkan oleh pemotongan.

Motor DC yang digunakan adalah motor DC 24V, menggunakan mikrokontroler atmega328 sebagai kontrolernya. Sistem bekerja sesuai spesifikasi yang diinginkan. Setelah dilakukan pengujian *open loop* dan dilakukan analisis penentuan kontroler maka di dapatkan  $K_p = 3,4$ ,  $K_i = 5,67$  dan  $K_d = 0,51$ . Dari hasil pengujian secara *closed loop* diperoleh *Time overshoot* sebesar 0,6 detik, *Settling Time* sebesar 0,45 detik, *Steady State* sebesar 0,9 detik, *Overshoot* 5,7%, *Error* 0,525%, *Recovery Time* sebesar 400mS. *Error* yang terjadi masih mampu membuat alat menghasilkan pemotongan yang baik.

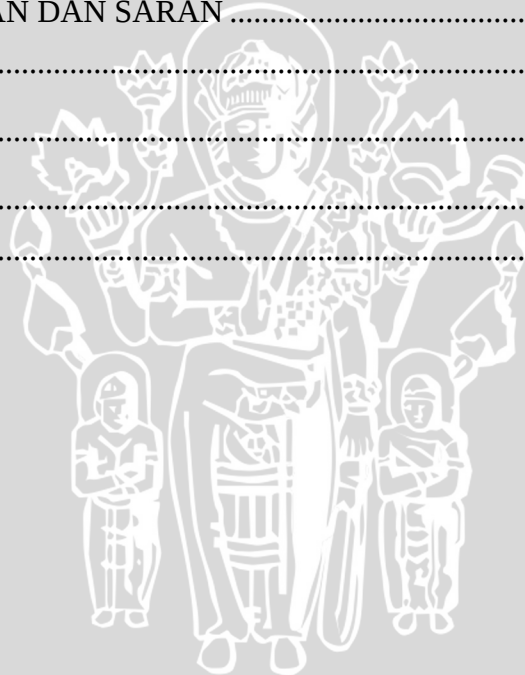
**Kata kunci** : Motor DC, Kecepatan Putaran, *Styrofoam*, *Optocoupler*

## DAFTAR ISI

|                                                             |     |
|-------------------------------------------------------------|-----|
| PENGANTAR .....                                             | iv  |
| ABSTRAK .....                                               | vii |
| DAFTAR ISI .....                                            | vii |
| DAFTAR GAMBAR .....                                         | x   |
| DAFTAR TABEL .....                                          | xi  |
| BAB I PENDAHULUAN .....                                     | 1   |
| 1.1 Latar Belakang .....                                    | 1   |
| 1.2 Rumusan Masalah .....                                   | 4   |
| 1.3 Batasan Masalah .....                                   | 4   |
| 1.4 Tujuan .....                                            | 5   |
| 1.5 Sistematika Penulisan .....                             | 5   |
| BAB II TINJAUAN PUSTAKA .....                               | 7   |
| 2.1 <i>Styrofoam</i> .....                                  | 7   |
| 2.1.1 Bahan Dasar <i>Styrofoam</i> .....                    | 7   |
| 2.1.2 Titik Leleh .....                                     | 8   |
| 2.1.3 Kekuatan dan Kelenturan .....                         | 8   |
| 2.1.4 Jenis-jenis <i>Styrofoam</i> .....                    | 9   |
| 2.2 Perkembangan Alat pemotong <i>Styrofoam</i> .....       | 11  |
| 2.3 Kontroler .....                                         | 13  |
| 2.4 Kontroler PID (Proporsional Integral Diferensial) ..... | 14  |
| 2.4.1 Kontroler Proporsional .....                          | 14  |
| 2.4.2 Kontroler Integral .....                              | 15  |
| 2.4.3 Kontroler Diferensial .....                           | 16  |
| 2.4.4. Kontroler Proporsional Integral Diferensial .....    | 17  |
| 2.5 Hand Tuning Kontroler PID .....                         | 19  |
| 2.6 PWM ( <i>Pulse Width Modulation</i> ) .....             | 20  |
| 2.7 Motor DC .....                                          | 21  |
| 2.8 Mikrokontroler .....                                    | 22  |

|                                             |                                                                |    |
|---------------------------------------------|----------------------------------------------------------------|----|
| 2.8.1                                       | Mikrokontroler ATmega328.....                                  | 23 |
| 2.9                                         | Arduino Uno.....                                               | 24 |
| 2.10                                        | Sensor <i>Optocoupler</i> .....                                | 25 |
| BAB III METODOLOGI PENELITIAN.....          |                                                                | 28 |
| 3.1                                         | Studi Literatur.....                                           | 28 |
| 3.2                                         | Perancangan Alat.....                                          | 29 |
| 3.2.1                                       | Perancangan Perangkat Keras dan Realisasi Pembuatan Alat ..... | 30 |
| 3.2.2                                       | Perancangan Perangkat Lunak .....                              | 30 |
| 3.3                                         | Pengujian Alat .....                                           | 30 |
| 3.3.1                                       | Pengujian Perangkat keras .....                                | 30 |
| 3.4.1                                       | Pengujian Perangkat Lunak.....                                 | 30 |
| 3.5                                         | Pengambilan Kesimpulan dan Saran .....                         | 31 |
| BAB IV PERANCANGAN DAN PEMBUATAN ALAT ..... |                                                                | 32 |
| 4.1                                         | Perancangan Sistem Mekanik.....                                | 32 |
| 4.2                                         | Spesifikasi Sistem Kontroler .....                             | 33 |
| 4.3                                         | Diagram Blok Sistem .....                                      | 34 |
| 4.4                                         | Perancangan Perangkat Keras .....                              | 36 |
| 4.4.1                                       | Motor DC .....                                                 | 36 |
| 4.4.2                                       | <i>Driver</i> Motor DC .....                                   | 37 |
| 4.4.3                                       | Sensor <i>Optocoupler</i> .....                                | 38 |
| 4.4.4                                       | Rangkaian <i>Frequency to Voltage</i> .....                    | 40 |
| 4.4.5                                       | Penentuan Nilai Penguatan Kontroler .....                      | 42 |
| 4.4.6                                       | Arduino Uno R3.....                                            | 42 |
| 4.5                                         | Perancangan Perangkat Lunak .....                              | 44 |
| BAB V PENGUJIAN DAN ANALISIS SISTEM .....   |                                                                | 45 |
| 5.1                                         | Pengujian sensor kecepatan putar motor DC .....                | 45 |
| 5.1.1                                       | Tujuan .....                                                   | 45 |
| 5.1.2                                       | Peralatan yang digunakan .....                                 | 45 |
| 5.1.3                                       | Langkah Pengujian.....                                         | 45 |
| 5.1.4                                       | Hasil Pengujian dan analisis.....                              | 46 |

|                                   |                                                                      |    |
|-----------------------------------|----------------------------------------------------------------------|----|
| 5.2                               | Pengujian respon sistem secara <i>open loop</i> .....                | 49 |
| 5.2.1                             | Tujuan .....                                                         | 49 |
| 5.2.2                             | Peralatan yang digunakan .....                                       | 49 |
| 5.2.3                             | Langkah Pengujian.....                                               | 49 |
| 5.2.4                             | Hasil Pengujian dan analisis.....                                    | 50 |
| 5.3                               | Analisis penentuan konstanta kontroler $K_p$ , $K_d$ dan $K_i$ ..... | 50 |
| 5.4                               | Pengujian respon sistem secara <i>closed loop</i> .....              | 52 |
| 5.4.1                             | Tujuan .....                                                         | 52 |
| 5.4.2                             | Peralatan yang digunakan .....                                       | 52 |
| 5.4.3                             | Langkah Pengujian.....                                               | 52 |
| 5.4.4                             | Hasil Pengujian dan analisis.....                                    | 52 |
| BAB VI KESIMPULAN DAN SARAN ..... |                                                                      | 54 |
| 6.1                               | Kesimpulan.....                                                      | 54 |
| 6.2                               | Saran .....                                                          | 54 |
| DAFTAR PUSTAKA .....              |                                                                      | 55 |
| LAMPIRAN .....                    |                                                                      | 56 |



## DAFTAR GAMBAR

|                                                                                                                               |    |
|-------------------------------------------------------------------------------------------------------------------------------|----|
| Gambar 2. 1 Cutter .....                                                                                                      | 11 |
| Gambar 2. 2 Hot wire .....                                                                                                    | 12 |
| Gambar 2. 3 Diagram Blok Kontroler Proporsional .....                                                                         | 14 |
| Gambar 2. 4 Diagram Blok Kontroler Integral .....                                                                             | 15 |
| Gambar 2. 5 Diagram Blok Kontroler Diferensial .....                                                                          | 16 |
| Gambar 2. 6 Diagram Blok Kontroler PID .....                                                                                  | 18 |
| Gambar 2. 7 Fungsi Waktu antara Sinyal Keluaran dan Sinyal Masukan Kontroler<br>Proporsional Integral Diferensial (PID) ..... | 18 |
| Gambar 2. 8 Gambar Sinyal PWM Secara Umum .....                                                                               | 20 |
| Gambar 2. 9 Motor DC .....                                                                                                    | 21 |
| Gambar 2. 10 Konfigurasi Pin ATmega 168-20AU .....                                                                            | 24 |
| Gambar 2. 11 Board Arduino Uno .....                                                                                          | 25 |
| Gambar 2. 12 Beberapa contoh detektor optik aplikasi elektronika daya .....                                                   | 27 |
| Gambar 4. 1 Sistem mekanik <i>hot-wire cutting</i> .....                                                                      | 33 |
| Gambar 4. 2 Diagram Blok Sistem .....                                                                                         | 35 |
| Gambar 4. 3 Motor DC .....                                                                                                    | 37 |
| Gambar 4. 4 Rangkaian driver motor DC .....                                                                                   | 38 |
| Gambar 4. 5 Beberapa contoh detektor optik aplikasi elektronika daya .....                                                    | 40 |
| Gambar 4. 6 Skematik Rangkaian Frequency to Voltage .....                                                                     | 41 |
| Gambar 4. 7 Board Arduino Uno R3 .....                                                                                        | 43 |
| Gambar 4. 8 Flowchart Perangkat Lunak .....                                                                                   | 44 |
| Gambar 5. 1 Hasil pengujian respon open loop motor DC .....                                                                   | 50 |
| Gambar 5. 2 Metode 1 Ziegler-Nichols (hasil pengujian) .....                                                                  | 51 |
| Gambar 5. 3 Hasil pengujian respon sistem secara closed loop .....                                                            | 53 |

## DAFTAR TABEL

|                                                                  |    |
|------------------------------------------------------------------|----|
| Tabel 5. 1 Hasil Pengujian sensor kecepatan putar motor DC.....  | 46 |
| Tabel 5. 2 Hasil konversi RPM.....                               | 48 |
| Tabel 5. 3 Aturan Metode 1 Ziegler-Nichols (Ogata K., 1997)..... | 51 |



## BAB I

### PENDAHULUAN

#### 1.1 Latar Belakang

Pada jaman sekarang plastik merupakan bahan yang sering dijumpai dan sering digunakan dalam kehidupan sehari-hari. Penggunaan plastik banyak digunakan pada perabotan rumah tangga sebagai *chasing* alat-alat elektronik dan lain sebagainya. Selain memiliki sifat yang tidak mudah pecah dan lentur, plastik juga memiliki nilai ekonomis yang tinggi dan mudah diaplikasikan sehingga penggunaannya sangat luas diberbagai bidang. Salah satu jenis plastik yang sering kita jumpai adalah *styrofoam* atau plastik busa. Istilah "*styrofoam*" sebenarnya merupakan nama dagang yang telah dipatenkan oleh perusahaan *dow chemical* untuk salah satu jenis plastik kemasan, yaitu *polystyrene foam*. *Styrofoam* sangat ringan, karena kandungan di dalamnya terdiri dari dua komposisi yaitu: 90-95% udara dan 5-10% *styrene*. Pada umumnya digunakan sebagai bahan pelindung dan penahan getaran barang-barang yang mudah pecah atau peka terhadap terhadap guncangan, seperti alat elektronik, jenis plastik ini dinamakan "gabus" oleh beberapa kalangan masyarakat di indonesia. Bahan dasar *styrofoam* adalah *polystyrene*, suatu jenis plastik yang sangat ringan, kaku, tembus cahaya, dan murah. Namun, bahan tersebut cepat rapuh. Untuk menutupi kelemahan tersebut, *polystyrene* dicampur seng dan senyawa *butadien*. Hal ini menyebabkan *polystyrene* kehilangan sifat jernihnya dan berubah menjadi putih susu.

Karena *styrofoam* banyak digunakan dalam banyak aplikasi dan kebutuhan akan bentuk tertentu maka diperlukan suatu teknik pemotongan yang bagus.

Teknik pemotongan *styrofoam* pada awalnya hanya menggunakan pisau biasa. Namun, ketika ditemukan pisau lipat atau yang saat ini populer disebut *cutter*, maka pisau biasa pun berangsur-angsur ditinggalkan dan jarang digunakan untuk memotong *syrofoam*. Hal ini disebabkan karena *cutter* ternyata memiliki banyak keunggulan, misalnya lebih ringan, lebih tajam dalam memotong, berkas potongannya sangat tipis dan halus. Seiring berkembangnya seni rupa dan dekorasi, maka kemampuan *cutter* sebagai pemotong *styrofoam* mulai ada kekurangannya. Terutama untuk memotong garis melengkung atau bentuk kurva *non-linear*. Sehingga diproduksi jenis pemotong *styrofoam* yang lebih *modern*, yaitu menggunakan elemen kawat yang dipanaskan menggunakan *heater* listrik. Hal ini dilatar belakangi dari sifat *styrofoam* yang sama seperti plastik lainnya yaitu dapat meleleh apabila dikenai panas. Alat ini hanya mampu digunakan untuk membentuk pola-pola berukuran kecil. Alat ini masih dioperasikan secara manual (menggunakan tangan). Alat ini sangat tergantung dengan keahlian tangan yang mengoprasikan. Selain itu alat ini menghasilkan panas yang cukup berbahaya dan dapat menimbulkan luka bakar yang serius jika tidak dioperasikan dengan baik.

*Styrofoam* sekarang juga digunakan sebagai bahan untuk membuat model pesawat *RC model* yang digunakan pada *aeromodelling*. Misalnya dalam pembuatan bagian badan pesawat (*fuselage*), sayap (*wing*) dan bagian-bagian pesawat lainnya. Hal ini dikarenakan sifat dari *styrofoam* yang ringan namun kuat, kokoh, mudah dibentuk dan lentur. Jika menggunakan aluminium, terlalu berat dan mahal dan jika menggunakan kayu maka kayu rawan patah, mudah lapuk serta rawan dimakan rayap walau mudah dipahat. Plastik jenis lainnya cukup bagus sebagai bahan dasar pembuatan pesawat *RC model*, namun plastik agak

sulit dibentuk dan harus dicetak terlebih dahulu dengan menggunakan cetakan. Dalam teknologi pembuatan pesawat *RC model* berbahan *styrofoam*, diperlukan teknik pemotongan *styrofoam* menggunakan kawat panas (*hot-wire cutting*) untuk masing-masing bagian. Untuk melakukan pemotongan bagian pesawat yang panjang, sebagai contoh bagian sayap atau badan pesawat diperlukan suhu *hot-wire* dengan kecepatan gerak *hot-wire* yang tepat. Umumnya pada pesawat *RC model* terdiri dari mesin, baterai, rangkaian *receiver*, *ESC* (*Electronic Speed Controller*) dsb, sehingga diperlukan ruang untuk menempatkan bagian-bagian tersebut agar rapi dan terlindung. Untuk membentuk bagian yang miring diperlukan pemotong yang dapat membentuknya secara simetris dan tidak simetris. Untuk menggerakkan *hot-wire* yang panjang untuk membentuk sayap pesawat dibutuhkan suatu mesin yang memiliki kecepatan stabil sesuai dengan suhu dan luas penampang *styrofoam* sehingga hasil pemotongannya menjadi lebih bagus.

*Fuselage* dibuat simetris dan rapi agar badan pesawat memiliki tingkat aerodinamis yang tinggi disamping untuk menempatkan titik keseimbangan badan pesawat *fuselage* tepat pada porosnya. Agar pemotongan dapat dilakukan cukup rapi dan simetris perlu adanya suatu *hot-wire* yang dapat memotong *styrofoam* yang tegak lurus atau mendatar, kedua ujung *hot-wire* ditahan oleh mekanikal yang dapat bergerak secara horizontal atau vertikal. Untuk menghasilkan pemotongan yang rapi perlu diatur kecepatan gerak dikedua sisi penahan *hot-wire*. Ketepatan pemotongan *styrofoam* ini sangat diperlukan agar kualitas pemotongan menjadi lebih baik dan memiliki ukuran yang tepat. Panas yang terlalu tinggi menyebabkan kualitas pemotongan menjadi tidak rata dan ukuran menjadi tidak

tepat. Hal ini tergantung pada jenis *styrofoam* yang akan dipotong karena titik leleh masing-masing *styrofoam* berbeda. Adapun parameter yang menentukan kualitas pemotongan adalah :

1. Temperatur kawat pemotong
2. Jenis *styrofoam*
3. Kecepatan pemotongan

*Disturbance* dalam sistem pengontrolan kecepatan pemotongan *styrofoam* secara *linear* ini adalah jenis *styrofoam* dan dimensi *styrofoam* yang akan dipotong. Dalam skripsi ini akan didesain sistem kontrol yang akan mengatur kecepatan gerak dikedua sisi penahan *hot-wire* dari bahan *nikeline*.

### 1.2 Rumusan Masalah

1. Bagaimana merancang kontroler PID (Proporsional Integral Diferensial) untuk mengendalikan kecepatan putaran motor DC pada pemotongan *styrofoam* secara *linear* ?
2. Bagaimana merancang *hardware* dan *software* sistem pengendalian kecepatan putaran motor DC agar hasil pemotongan lebih rapi ?

### 1.3 Batasan Masalah

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

1. Bagaimana mengontrol kecepatan putar dari motor DC agar didapatkan kecepatan yang sama di kedua sisinya.
2. Menggunakan motor DC 24V sebagai aktuator.
3. Tidak membahas modul *hot wire*.

4. Pengendali PID diterapkan menggunakan mikrokontroler ATmega328.
5. Tidak membahas catu daya sistem alat pemotongan *styrofoam*.
6. Desain mekanik dan elektronika tidak dibahas mendalam.

#### 1.4 Tujuan

Tujuan yang ingin dicapai dari tugas akhir ini adalah mengatur kecepatan gerak pemotongan dikedua sisi dengan menggunakan kontroler PID berbasis mikrokontroler ATmega 328.

#### 1.5 Sistematika Penulisan

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

##### BAB I

##### Pendahuluan

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

##### BAB II

##### Tinjauan Pustaka

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

##### BAB III

##### Metodologi Penelitian

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

**BAB IV Perancangan dan Pembuatan Alat**

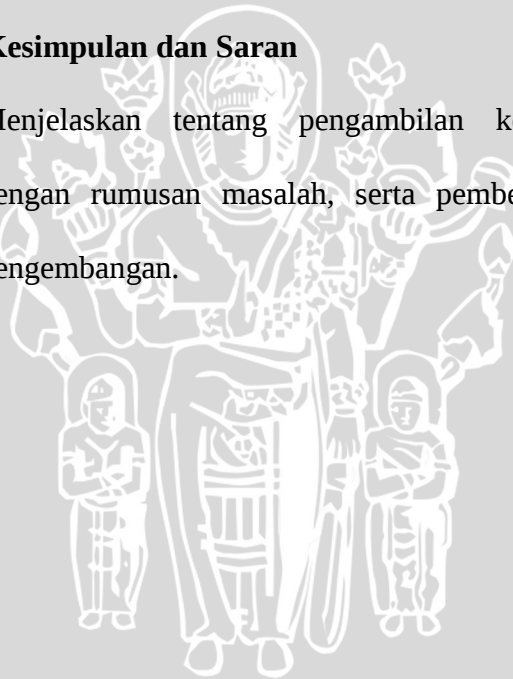
Menjelaskan tentang perancangan dan pembuatan alat yang meliputi prinsip kerja alat, perancangan perangkat keras, dan perangkat kontroler PID dan menerapkannya ke dalam *software*, sehingga sistem dapat bekerja dengan baik.

**BAB V Pengujian dan Analisis**

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

**BAB VI Kesimpulan dan Saran**

Menjelaskan tentang pengambilan kesimpulan sesuai dengan rumusan masalah, serta pemberian saran untuk pengembangan.



## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Styrofoam

*Styrofoam* atau plastik busa adalah salah satu jenis palastik. Kata "*styrofoam*" sebenarnya merupakan nama dagang yang telah dipatenkan oleh perusahaan *Dow Chemical* untuk salah satu jenis plastik kemasan, yaitu *polystyrene foam*. *Polistirena foam* dikenal luas dengan istilah *styrofoam* yang seringkali digunakan secara tidak tepat oleh publik karena sebenarnya *styrofoam* merupakan nama dagang yang telah dipatenkan oleh perusahaan *Dow Chemical*. Oleh pembuatnya *styrofoam* dimaksudkan untuk digunakan sebagai insulator pada bahan konstruksi bangunan, bukan untuk kemasan pangan. 5-10% gas seperti *n-butana* atau *n-pentana*. Polistirena dibuat dari monomer stirena melalui proses polimerisasi. *Polistirena foam* dibuat dari monomer stirena melalui polimerisasi suspensi pada tekanan dan suhu tertentu, selanjutnya dilakukan pemanasan untuk melunakkan resin dan menguapkan sisa *blowing agent*. *Polistirena foam* merupakan bahan plastik yang memiliki sifat khusus dengan struktur yang tersusun dari butiran dengan kerapatan rendah, mempunyai bobot ringan, dan terdapat ruang antar butiran yang berisi udara. Kemasan plastik jenis *polistirena* sering menimbulkan masalah pada lingkungan karena bahan ini sulit mengalami peruraian biologik dan sulit didaur ulang.

##### 2.1.1 Bahan Dasar Styrofoam

*Styrofoam* sangat ringan, karena kandungan di dalamnya 90-95% udara dan 5-10% *styrene*. Umumnya digunakan sebagai bahan pelindung dan penahan

getaran barang-barang yang mudah pecah atau peka terhadap goncangan, seperti alat elektronik, jenis plastik ini dinamakan "gabus" oleh beberapa kalangan masyarakat Indonesia. Bahan dasar *styrofoam* adalah *polystyrene*, suatu jenis plastik yang sangat ringan, kaku, tembus cahaya, dan murah. Namun, bahan tersebut cepat rapuh. Karena kelemahannya tersebut, *polystyrene* dicampur dengan senyawa *butadien*. Hal ini menyebabkan *polystyrene* kehilangan sifat jernihnya dan berubah menjadi putih susu. Kemudian untuk kelenturannya, ditambahkan zat plasticier seperti dioktiltatarat (DOP), *butil hidroksi tolulena*, atau *butyl stearat*. Plastik busa yang mudah terurai menjadi struktur sel-sel kecil merupakan hasil proses peniupan dengan menggunakan gas *chlorofluorocarbon* (CFC). Hasilnya adalah bentuk seperti yang dipergunakan saat ini. *Polystyrene* memiliki massa jenis sampai  $1050 \text{ kg/m}^3$ , kuat tarik sampai  $40 \text{ MN/m}^2$ , modulus geser sampai  $0,990 \text{ GN/m}^2$  dan angka *poisson* 0,330 (Crawford, 1998 dalam Wijaya, 2005).

### 2.1.2 Titik Leleh

Karena merupakan salah satu jenis plastik yang lentur, maka *styrofoam* juga dapat meleleh jika dipanaskan. Umumnya *polystyrene* yang merupakan penyusunan utama *styrofoam* meleleh pada suhu  $80^\circ\text{C}$ . ("Bahaya dan Keuntungan Kemasan Primer" 2008, July).

### 2.1.3 Kekuatan dan Kelenturan

Pada dasarnya *styrofoam* merupakan bahan cepat rapuh. Karena kelemahannya tersebut, *polystyrene* dicampur dengan senyawa *butadien*. Hal ini menyebabkan *polystyrene* kehilangan sifat jernihnya dan berubah menjadi putih susu. Kemudian untuk kelenturannya, ditambahkan zat plasticier seperti

*dioktilptatat* (DOP), *butil hidroksi tolulena*, atau *n-butyl stearat* plastik busa yang mudah terurai menjadi struktur sel-sel kecil merupakan hasil proses peniupan dengan menggunakan gas *chorofluorocarbon* (CFC). Hasilnya adalah bentuk seperti yang dipergunakan saat ini *polyster* memiliki massa jenis sampai 1050 kg/m<sup>3</sup>, kuat tarik sampai 40 MN/m<sup>2</sup>, modulus geser sampai 0,990 GN/ m<sup>2</sup> dan angka *poisson* 0,330(crawford,1998 dalam Wijaya, 2005). Koefisien gesek *polystyrene* atau *styrofoam* dengan baja, yaitu antara 0,3-0,35 (coefficient of friction, 2006). Karena merupakan salah satu jenis plastik yang lentur, maka *styrofoam* pun dapat meleleh jika dipanaskan. Umumnya *polystyrene* yang merupakan penyusunan utama *styrofoam* meleleh pada suhu 80°C.(“bahaya dan keuntungan kemasan primer”2008, july).

#### 2.1.4 Jenis-jenis *Styrofoam*

##### 2.1.4.1 EPO

EPO (*Piocelan*) adalah bahan busa baru yang merupakan campuran dari PE dan PS. bahan EPO mempunyai daya tahan tertinggi, serta berkualitas baik dan jenis busa yang relatif tangguh dan tidak patah seperti beberapa jenis busa lainnya.

##### 2.1.4.1 EPS

*Expanded polystyrene* (EPS) busa adalah busa paling umum dipotong untuk bentuk arsitektur, sistem panel struktural terisolasi (SIPS), terisolasi bentuk beton (ICF), cetakan beton pracetak dan model dengan tema, dan tanda pembuatan. EPS secara teknis plastik. plastik 'Kata' berasal dari *plastikos* Yunani,

yang berarti mampu menjadi dibentuk. Kebebasan untuk berkreasi dengan desain khusus adalah alasan utama mengapa kita menemukan EPS hari ini begitu berharga. Hal ini juga menarik untuk mengetahui bahwa busa EPS adalah bahan yang sama (*polystyrene*) digunakan untuk membuat kasus CD cangkang keras dan produk rumah tangga lainnya. Untuk menemukan polistiren dalam produk sehari-hari, mencari daur ulang.

#### 2.1.4.2 EPP

*Poliolefin* - *poliolefin* adalah kelompok terbesar termoplastik, sering disebut sebagai termoplastik komoditi, mereka adalah polimer olefin sederhana seperti *ethylene*, *propylene*, *butenes*, *isoprenes*, dan *pentenes*, dan kopolimer dan modifikasi daripadanya. *Poliolefin* hanya terdiri dari atom karbon dan hidrogen dan mereka adalah non-aromatik. *Poliolefin* biasanya diproses oleh *ekstrusi*, *injection molding*, *blow molding*, dan *metode molding rotasi*. *Thermoforming*, *calender*, dan pencetakan kompresi digunakan untuk tingkat yang lebih rendah. Sebuah karakteristik yang melekat umum untuk semua *poliolefin* adalah, nonpolar tidak keropos, rendah energi permukaan yang tidak menerima tinta, dan lak tanpa *pretreatment* oksidatif khusus. Dua *poliolefin* paling penting dan umum adalah *polyethylene* dan *polypropylene* (EPP) mereka sangat populer karena biaya rendah dan berbagai aplikasi.

#### 2.1.4.3 Depron

Depron adalah sel busa *styrofoam* tertutup dalam bentuk lembaran, merupakan media yang bagus untuk membangun pesawat *model*, dan juga media yang populer untuk membangun suatu model arsitektur, serta perahu *model* dan desain *prototype*. Bahan itu cukup ringan dan kuat untuk membawa motor baterai

elektrik yang ringan. Depron dikembangkan terutama digunakan untuk sebagai isolasi panas dan suara. Depron sebenarnya *polystyrene* (EPS/XPS) yang lebih kuat dan padat yang terjadi dalam bentuk sel busa yang tertutup. Buka atau tertutupnya sel mengacu pada struktur dank arena sel jenis terbuka (jenis spons) dapat menyerap air atau lem sehingga sel jenis terbuka kurang cocok untuk pembuatan pesawat model. Sel busa tertutup lebih unggul karena lebih sedikit menyerap lem dan lebih tahan air. Permukaanya yang tertutup menambahkan kekuatan yang cukup serta sangat mudah untuk mengaplikasikan cat tidak larut depiron yang ramah atau tinta sebagai *finishing* pesawat *model*.

## 2.2 Perkembangan Alat pemotong *Styrofoam*

Alat pemotong *styrofoam* pada awalnya hanya menggunakan pisau biasa. Namun, ketika ditemukan pisau lipat atau yang saat ini populer disebut *cutter*, maka pisau biasa pun berangsur-angsur ditinggalkan dan jarang digunakan untuk memotong *styrofoam*. Hal ini dipicu karena *cutter* ternyata memiliki banyak keunggulan, misalnya lebih ringan, lebih tajam dalam memotong, berkas potonganya sangat tipis dan halus.



Gambar 2. 1 Cutter

Sumber : <http://www.okokchina.com>

Seiring berkembangnya seni rupa dan dekorasi, maka kemampuan *cutter* sebagai pemotong *styrofoam* mulai ada kekurangannya. Terutama untuk memotong garis melengkung atau bentuk kurva *non linear*. Sehingga diproduksi jenis pemotong *styrofoam* yang lebih *modern*, yaitu menggunakan elemen kawat yang dipanaskan menggunakan *heater* listrik. Hal ini dilatar belakangi dari sifat *styrofoam* yang meleleh apabila dikenai panas. Alat ini hanya mampu digunakan untuk membentuk pola-pola berukuran kecil. Alat ini masih dioperasikan secara manual(menggunakan tangan). Alat ini sangat tergantung dengan keahlian tangan yang mengoperasikan. Selain itu,alat ini menghasilkan panas yang cukup berbahaya dan dapat menimbulkan luka bakar yang serius jika tidak dioperasikan dengan baik.



**Gambar 2. 2 Hot wire**

Sumber : <http://www.anticsonline.co.uk>

*Styrofoam* sekarang juga digunakan untuk membuat model pesawat *RC model* yang digunakan pada *aeromodeling*. Misalnya dalam pembuatan *fuselage* (bagian badan pesawat), *wing* (sayap) dan bagian-bagian pesawat lainnya. Hal ini

dikarenakan sifat dari *styrofoam* yang ringan namun kuat dan kokoh, mudah dibentuk dan lentur. Jika menggunakan aluminium, terlalu berat serta sulit dibentuk dan jika menggunakan kayu maka kayu rawan patah, mudah lapuk serta rawan dimakan rayap walau mudah dipahat. Plastik cukup bagus sebagai bahan dasar pembuatan pesawat *RC model*, namun plastik agak sulit dibentuk dan harus dicetak terlebih dahulu.

### 2.3 Kontroler

Dengan adanya kontroler dalam sebuah sistem kontrol sangat berperan penting terhadap seluruh perilaku yang terjadi pada sistem. Pada dasarnya semua itu disebabkan oleh komponen yang digunakan sebagai perancangan *system* tersebut. Artinya, karakteristik *plant* yang digunakan harus dapat diterima sebagaimana adanya, sehingga segala pergerakan dari sistem hanya dapat dilakukan dengan menambahkan subsistem yaitu kontroler.

Salah satu fungsi komponen kontroler adalah mengurangi sinyal kesalahan atau tingkat kesalahan sistem, yaitu perbedaan antara nilai referensi/nilai yang diinginkan dan nilai aktual. Dengan cara tersebut akan sesuai dengan tujuan sistem kontrol di mana mendapat nilai aktual atau sinyal keluaran sama dengan nilai yang diinginkan/referensi. Semakin kecil kesalahan yang terjadi, semakin baik kinerja sistem kontrol yang diterapkan.

Jika perbedaan antara nilai referensi dengan nilai keluaran selisihnya relatif besar, maka kontroler yang baik harus dapat mempengaruhi *plant* agar memperkecil selisih nilai keluaran *plant* dengan nilai referensi sekecil mungkin secara cepat dan tepat.

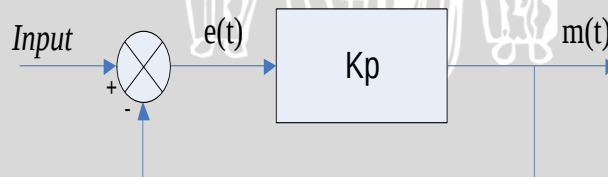
Prinsip kerja kontroler adalah membandingkan nilai aktual keluaran *plant* dengan nilai referensi, kemudian menentukan nilai kesalahan dan akhirnya menghasilkan sinyal kontrol untuk meminimalkan kesalahan (Ogata, K., 1997).

## 2.4 Kontroler PID (Proporsional Integral Diferensial)

### 2.4.1 Kontroler Proporsional

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

Pada Gambar 2.3 menunjukkan diagram blok yang menggambarkan hubungan antara *input* (besaran referensi yang diinginkan), besaran aktual dengan besaran keluaran kontroler proporsional, dan besaran kesalahan (*error*). Sinyal kesalahan (*error*) merupakan selisih antara besaran *setting* dengan besaran aktualnya.



Gambar 2. 3 Diagram Blok Kontroler Proporsional

Sumber: Ogata, K., 1997: 157

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

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

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

## 2.4.2 Kontroler Integral

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

Kontroler integral memiliki karakteristik seperti halnya sebuah integral. Keluaran kontroler sangat dipengaruhi oleh perubahan yang sebanding dengan nilai sinyal kesalahan. Keluaran kontroler ini merupakan jumlahan yang terus menerus dari perubahan masukannya. Kalau sinyal kesalahan tidak mengalami perubahan, keluaran akan menjaga keadaan seperti sebelum terjadinya perubahan masukan. Gambar 2.4 menunjukkan diagram blok kontroler integral.



**Gambar 2. 4** Diagram Blok Kontroler Integral

Sumber: Ogata, K., 1997: 158

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

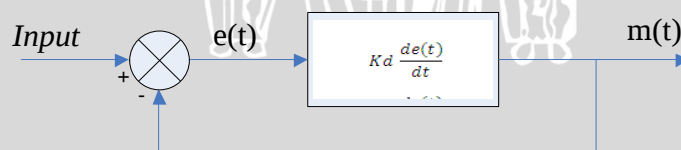
$$\frac{dm(t)}{dt} = K_i \cdot e(t) \dots\dots\dots(2-2)$$

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

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

### 2.4.3 Kontroler Diferensial

Kontroler Diferensial memiliki sifat seperti halnya suatu operasi derivatif. Perubahan yang mendadak pada masukan kontroler, akan mengakibatkan perubahan yang sangat besar dan cepat. Gambar 2.5 berikut menunjukkan diagram blok pada kontroler Diferensial.



**Gambar 2. 5** Diagram Blok Kontroler Diferensial

Sumber: Ogata, K., 1997: 177

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

Hubungan ini dapat ditulis sebagai:

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

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

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

Dari ketiga aksi kontrol dasar di atas dapat dibuat kombinasi dari ketiganya, yaitu kontroler Proporsional Integral Diferensial (PID)

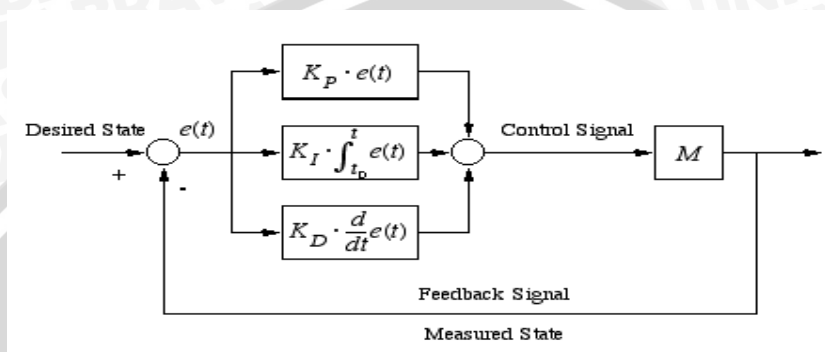
#### 2.4.4. Kontroler Proporsional Integral Diferensial

Setiap kekurangan dan kelebihan dari masing-masing kontroler Proporsional(P), Integral(I) dan Diferensial (D) dapat saling menutupi dengan menggabungkan ketiganya secara paralel menjadi kontroler proporsional integral Diferensial (PID). Elemen-elemen kontroler Proporsional(P), Integral(I) dan Diferensial (D) masing-masing secara keseluruhan bertujuan untuk mempercepat reaksi sebuah sistem, menghilangkan *offset* dan menghasilkan perubahan awal yang besar (Gunterus, 1994, 8-10). Kontroler Proporsional Integral Diferensial (PID) memiliki diagram kendali seperti dalam Gambar 2.6.

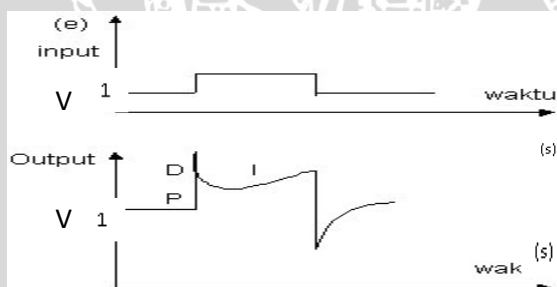
Aksi kontrolnya dinyatakan sebagai:

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

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



**Gambar 2. 6** Diagram Blok Kontroler PID



**Gambar 2. 7** Fungsi Waktu antara Sinyal Keluaran dan Sinyal Masukan Kontroler Proporsional Integral Diferensial (PID)

Sumber: Gunterus, 1994:8-11

Keluaran kontroler Proporsional Integral Diferensial (PID) merupakan penjumlahan dari keluaran kontroler proporsional, integral dan Diferensial.

Gambar 2.6 menunjukkan hubungan tersebut. Karakteristik kontroler Proporsional Integral Diferensial (PID) sangat dipengaruhi oleh kontribusi besar dari ketiga

parameter Proporsional(P),Integral(I) dan Diferensial(D). Penyetelan konstanta  $K_p$ ,  $T_i$  dan  $T_d$  akan mengakibatkan penonjolan sifat dari masing-masing elemen. Satu atau dua dari ketiga konstanta tersebut dapat disetel lebih menonjol dibanding yang lain.

Konstanta yang menonjol itulah yang akan memberikan kontribusi pada *respons* sistem secara keseluruhan (Gunterus, 1994, 8-10).

## 2.5 Hand Tuning Kontroler PID

Kontroler PID dapat di *tuning* dalam beberapa cara, antara lain Ziegler-Nichols *tuning*, *loop shaping*, metode analitis, optimisasi, *pole placement*, *auto tuning* dan *hand tuning* (Smith, 1979; Astrom & Hagglund, 1995).

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

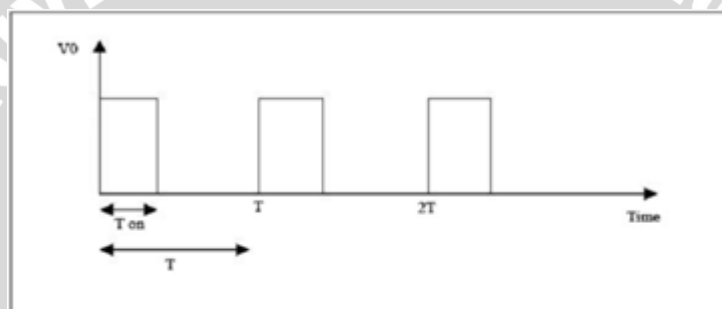
1. Melepaskan kontroler integral dan diferensial dengan memberikan nilai  $K_i = 0$  dan  $K_d = 0$ .
2. Mengatur nilai  $K_p$  hingga didapatkan respons yang diinginkan, dengan mengabaikan *offset* dari *setpoint*.
3. Dengan terus menaikkan nilai  $K_p$ , nilai dari  $K_d$  dinaikkan untuk mengurangi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

Keuntungan dari *hand tuning* adalah prosedur diatas dapat dilakukan dengan segera, *online* dan dapat melihat dengan cepat respons sistem setelah perubahan  $K_p$ ,  $K_i$  dan  $K_d$ . Kerugian dari cara ini adalah kesulitan untuk melihat apakah *setting* akhir dari kontroler merupakan nilai optimal atau tidak (Jantzen, 2001).

## 2.6 PWM (Pulse Width Modulation)

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

Pada sinyal PWM, frekuensi sinyal konstan sedangkan *duty cycle* bervariasi dari 0%-100%. Dengan mengatur *duty cycle* akan diperoleh keluaran yang diinginkan. Sinyal PWM (*Pulse Width Modulation*) secara umum dapat dilihat dalam Gambar 2.8 berikut:



**Gambar 2.8** Gambar Sinyal PWM Secara Umum

Sumber: electronics-scheme.com

$$Dutycycle = \frac{T_{on}}{T} \times 100\% \dots (\%) \dots \dots \dots (6-13)$$

Dengan:

$T_{on}$  = Periode logika tinggi

$T$  = Periode keseluruhan

$$V_{dc} = Dutycycle \times V_{cc} \dots (V) \dots \dots \dots (6-14)$$

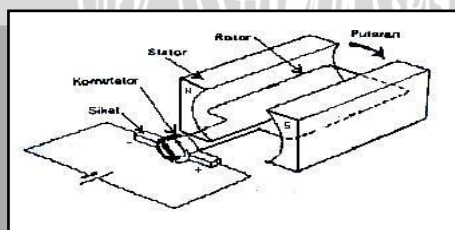
Sedangkan frekuensi sinyal dapat ditentukan dengan rumus berikut:

$$f_{0n} = \frac{f_{clk I/O}}{N.256} \dots (Hz) \dots \dots \dots (6-15)$$

## 2.7 Motor DC

Motor DC merupakan motor arus searah yaitu suatu mesin yang berfungsi mengubah tenaga listrik arus searah menjadi tenaga gerak atau tenaga mekanik, di mana gerak tersebut berupa putaran dari pada rotor.

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 2.9 di bawah ini:



**Gambar 2. 9** Motor DC

Sumber : Kismet F [1994:98]

## 2.8 Mikrokontroler

Mikrokontroler populer yang pertama dibuat oleh Intel pada tahun 1976, yaitu mikrokontroler 8-bit Intel 8748. Mikrokontroler tersebut adalah bagian dari keluarga mikrokontroler MCS-48. Sebelumnya, Texas instruments telah memasarkan mikrokontroler 4-bit pertama yaitu TMS 1000 pada tahun 1974. TMS 1000 yang mulai dibuat sejak 1971 adalah mikrokomputer dalam sebuah *chip*, lengkap dengan RAM dan ROM.

Pengendali mikro (*microcontroller*) adalah sistem mikroprosesor lengkap yang terkandung di dalam sebuah *chip*. Mikrokontroler berbeda dari mikroprosesor serba guna yang digunakan dalam sebuah PC, karena sebuah mikrokontroler umumnya telah berisi komponen pendukung sistem minimal mikroprosesor, yakni memori dan antarmuka I/O.

Berbeda dengan CPU serba-guna, mikrokontroler tidak selalu memerlukan memori eksternal, sehingga mikrokontroler dapat dibuat lebih murah dalam kemasan yang lebih kecil dengan jumlah *pin* yang lebih sedikit.

Sebuah *chip* mikrokontroler umumnya memiliki fitur:

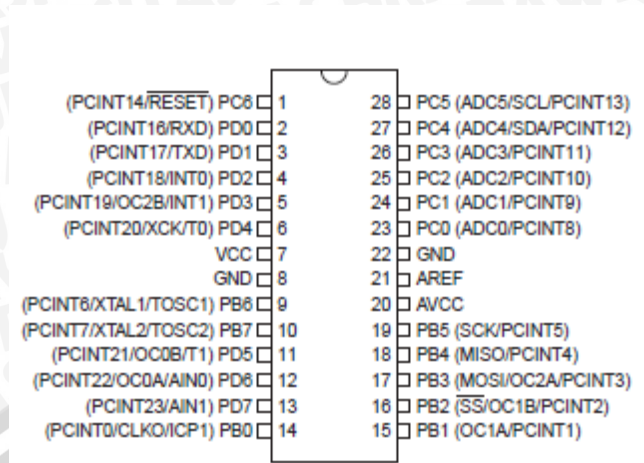
- a) *central processing unit* - mulai dari prosesor 4-bit yang sederhana hingga prosesor kinerja tinggi 64-bit.
- b) input/output antarmuka jaringan seperti *port serial* (UART)
- c) antarmuka komunikasi serial lain seperti I<sup>2</sup>C, *Serial Peripheral Interface* and *Controller Area Network* untuk sambungan sistem
- d) periferal seperti *timer* dan *watchdog*
- e) RAM untuk penyimpanan data

- f) ROM, EPROM, EEPROM atau *Flash memory* untuk menyimpan program komputer
- g) pembangkit *clock* - biasanya berupa resonator rangkaian RC
- h) pengubah analog-ke-digital

### 2.8.1 Mikrokontroler ATmega328

Atmel ATmega328 adalah mikrokontroler CMOS 8-bit berdaya rendah berbasis AVR yang arsitektur RISCnya telah ditingkatkan. Hampir semua instruksi dieksekusi dalam satu siklus *clock*, mempunyai *throughput* mendekati 1 MIPS per MHz membuat desainer sistem untuk mengoptimasi konsumsi daya versus kecepatan proses.

Atmel ATmega328 memiliki beberapa fitur antara lain 8Kbytes *In-system Programmable Flash with Read-While-Write*, 1K bytes EEPROM, 2K bytes SRAM, 23 jalur I/O untuk tujuan umum, 32 *working registers* untuk tujuan umum, tiga *timer/counter* yang fleksibel dengan *compare mode*, internal dan *external interrupt*, sebuah serial *programmable USART*, sebuah *byte-oriented 2-wireSerial Interface*, sebuah port SPI serial, sebuah 6-channel 10-bit ADC, sebuah *Watchdog Timer* yang *programmable* dengan internal osilator. Seperti dalam Gambar 2.10



Gambar 2. 10 Konfigurasi Pin ATmega 168-20AU

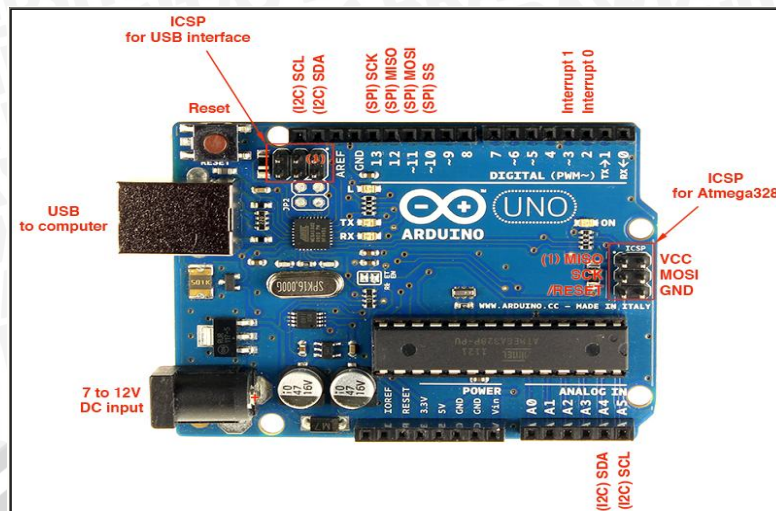
Sumber: [www.atmel.com](http://www.atmel.com)

## 2.9 Arduino Uno

Arduino Uno adalah sebuah *board* mikrokontroler berbasis ATmega328.

Arduino Uno mempunyai 14 pin *digital input/output* (6 di antaranya dapat digunakan sebagai *output* PWM), 6 input analog, sebuah osilator Kristal 16 MHz, sebuah koneksi USB, sebuah *power jack*, sebuah ICSP *header*, dan sebuah tombol *reset*. Arduino Uno memuat semua yang dibutuhkan untuk menunjang mikrokontroler, mudah menghubungkannya ke sebuah komputer dengan sebuah kabel USB atau mensuplainya dengan sebuah *adaptor* AC ke DC atau menggunakan baterai untuk memulainya.

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. 11 Board Arduino Uno

Sumber : [electroschematics.com](http://electroschematics.com)

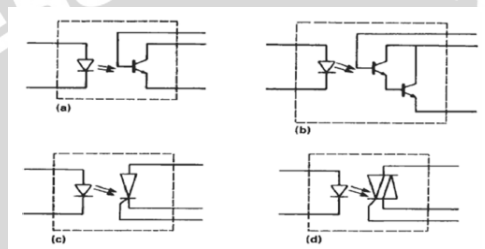
Nama “Uno” berarti *satu* dalam bahasa Italia, untuk menandai peluncuran Arduino 1.0. Uno dan versi 1.0 akan menjadi versi referensi dari Arduino. Uno adalah yang terbaru dalam serangkaian board USB Arduino, dan sebagai model referensi untuk *platform* Arduino, untuk perbandingan dengan versi sebelumnya.

## 2.10 Sensor Optocoupler

Komponen yang digunakan untuk mengisolasi komponen elektronika daya dengan rangkaian berdaya rendah yang sebagai rangkaian pengontrol salahsatunya adalah *optocoupler*. Sering kali ditemukan rangkaian elektronika daya orde megawatt dikontrol hanya dengan rangkaian beberapa watt. *Optocoupler* berfungsi sebagai pengisolasi sinyal listrik antara rangkaian sinyal masukan dan keluaran (Mazda, 1997: 82). *Optocoupler* merupakan alat yang digunakan untuk mengkopel suatu sumber dengan detektor menggunakan perantara cahaya. Oleh karena itu piranti ini disebut dengan *optoisolator* / *optocoupler*. *Optocoupler*

terbuat dari kombinasi dari sumber cahaya dan pendeteksi cahaya yang tergabung dalam satu kemasan. *Light emitting diode* atau *LED* seringkali digunakan sebagai sumber cahaya, dan berbagai komponen semikonduktor sebagai detektor cahaya. Kaca atau plastik digunakan sebagai pemisah antara sumber cahaya dan detektor, sehingga memungkinkan untuk ditempatkan pada tempat yang berdekatan. Pada prinsipnya, sinyal listrik dalam bentuk arus pada masukan diubah menjadi sinyal optik dengan menggunakan sumber cahaya LED. Sinyal optik tersebut akan diterima oleh detektor untuk diubah kembali menjadi sinyal listrik. Umumnya *optocoupler* dipakai untuk mengisolasi sinyal listrik yang ada pada rangkaian masukan dan keluaran sehingga dapat digunakan transmisi sinyal antar rangkaian. *Optocoupler* memiliki beberapa *parameter* antara lain: tingkat isolasi antarasumber cahaya dan detektor, rasio transfer arus masukan-luaran, dan kecepatan operasi *optocoupler*. Rasio *transfer* arus merupakan rasio antara arus keluaran dan arus masukan dari sumber potensial, ketika *optocoupler* bekerja. Berbagai jenis dari detektor yang digunakan pada *optocoupler* ditunjukkan pada Gambar 6.10 foto transistor-*coupler* dikenal dalam kategori harga rendah dengan kecepatan operasi kerja 100-500 kHz dan minimum rasio *transfer* arus antara 20% dan 300%. Foto darlington memiliki rasio *transfer* arus antara 100% dan 1000% namun memiliki tingkat keakuratan yang rendah dikarenakan adanya tingkat penguatan pada hubungan darlington dengan kecepatan operasi termasuk rendah, dengan nilai antara 20 kHz dan 10 kHz. Foto thyristor dan foto triac digunakan sebagai detector untuk aplikasi dengan arus yang besar. Namun *parameter* yang perlu diperhatikan lebih lanjut adalah arus yang mengalir melalui *LED* yang nantinya akan memicu thyristor atau TRIAC. Efisiensi hantaran antara *LED*

dengan foto transistor yang rendah sehingga perlu untuk mendesain thyristor atau TRIAC yang memiliki tingkat sensitivitas yang tinggi. Biasanya proses desain memerlukan perhatian lebih agar tidak menurunkan *parameter* yang lain seperti kemampuan hantaran arus. Beberapa *parameter* umum dari foto transistor dan foto triac antara lain : arus pemicuan 10-130 mA, mampu menyalurkan 100 sampai 300 mA, dan Waktu kerja 1-10 ms.



**Gambar 2. 12** Beberapa contoh detektor optik aplikasi elektronika daya.

a) Transistor, b) Darlington, c) Thyristor, d) Triac

Sumber : Mazda (1997:83)

### BAB III

#### METODOLOGI PENELITIAN

Kajian dalam skripsi ini merupakan penelitian yang bersifat aplikatif, yaitu merancang dan membuat suatu kontroler alat pemotong *styrofoam* dengan dua penggerak motor listrik yang menitik beratkan pada stabilisasi kecepatan putaran dengan menggunakan Kontroler (PID) Proporsional Integral Diferensial. Tujuannya agar dapat menampilkan performansi sistem sesuai dengan yang diinginkan.

Langkah-langkah yang perlu dilakukan untuk merealisasikan alat yang akan dibuat adalah sebagai berikut:

1. Studi literatur
2. Perancangan alat
3. Pengujian alat
4. Pengambilan kesimpulan

#### 3.1 Studi Literatur

Studi literatur mengacu pada prinsip kerja alat yang dirancang meliputi sistem Pengendalian Kecepatan Motor DC untuk Pemotongan Styrofoam Secara *Linear* dengan *Hotwire* Berbasis Mikrokontroler, karakteristik dari komponen yang digunakan serta studi tentang pengendalinya yaitu mikrokontroler atmega328.

Adapun hal-hal yang berhubungan dengan hal tersebut adalah :

1. Studi tentang teori dasar pemotongan *styrofoam* yaitu meliputi :  
Temperatur kawat pemotong, jenis *styrofoam*, kecepatan pemotongan.
2. Studi tentang rangkaian sensor, yaitu meliputi : Karakteristik spesifikasi sensor kecepatan *optocoupler*, penerapan sensor kecepatan sesuai sistem. karakteristik komponen elektronik yang digunakan diantaranya meliputi driver motor DC, Mikrokontroler atmega328, Komponen IRF640 sebagai saklar elektrik motor DC.
3. Mikrokontroler atmega328, yaitu meliputi : Karakteristik umum mikrokontroler atmega328, cara memprogram mikrokontroler atmega328 sebagai pusat pengontrol kecepatan putaran motor DC

### 3.2 Perancangan Alat

Perancangan sistem dilakukan sebagai langkah awal sebelum terbentuknya suatu sistem beserta rangkaian elektronik pendukungnya, hal ini dimaksudkan agar sistem pengendalian pemotongan *styrofoam* secara *linear* menggunakan motor DC tersebut dapat berjalan sesuai dengan deskripsi awal yang telah direncanakan. Perancangan sistem yang dilakukan meliputi:

1. Penentuan dimensi dan bahan yang digunakan dalam pembuatan alat pemotong *styrofoam* secara *linear*.
2. Penentuan rangkaian elektronik yang digunakan.
3. Perancangan program pada mikrokontroler ATmega328 sebagai pusat pengontrol.
4. Penentuan spesifikasi sistem yang akan dibuat, meliputi :
  - a. Penentuan deskripsi kerja sistem secara keseluruhan.

- b. Mikrokontroler ATmega328 dan rangkaian elektronik pendukung

### 3.2.1 Perancangan Perangkat Keras dan Realisasi Pembuatan Alat

- a. Pembuatan diagram blok
- b. Penentuan dan Perhitungan komponen yang akan digunakan dalam perancangan alat
- c. Merakit perangkat keras (*hardware*) untuk masing-masing blok.

### 3.2.2 Perancangan Perangkat Lunak

Perancangan perangkat lunak dimulai dari pembuatan *flowchart*, kemudian penulisan *listing code* program PID pada software Arduino ERW 1.0.5.

## 3.3 Pengujian Alat

Setelah semua komponen pada alat sudah terealisasi sesuai dengan diagram blok yang telah dirancang dan perangkat lunak pendukung sistem sudah dibuat, maka diadakan pengujian dan analisa alat.

### 3.3.1 Pengujian Perangkat keras

Pengujian perangkat keras dilakukan dengan tujuan untuk menyesuaikan nilai tegangan dan arus yang diijinkan bekerja dalam komponen berdasarkan data sekunder komponen yang diambil dari buku data komponen elektronika maupun dari *datasheet*.

### 3.4.1 Pengujian Perangkat Lunak

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

### 3.5 Pengambilan Kesimpulan dan Saran

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



## BAB IV

### PERANCANGAN DAN PEMBUATAN ALAT

Dalam bab ini membahas mengenai perancangan dan pembuatan sistem pengendali kecepatan putaran motor DC untuk pemotongan *Styrofoam* secara *linear* menggunakan kontroler PID. Perancangan perangkat tersebut meliputi perancangan perangkat keras maupun perancangan perangkat lunak. Sedangkan pembuatan bertujuan untuk menghasilkan semua perangkat pendukung maupun alat secara keseluruhan..

#### 4.1 Perancangan Sistem Mekanik

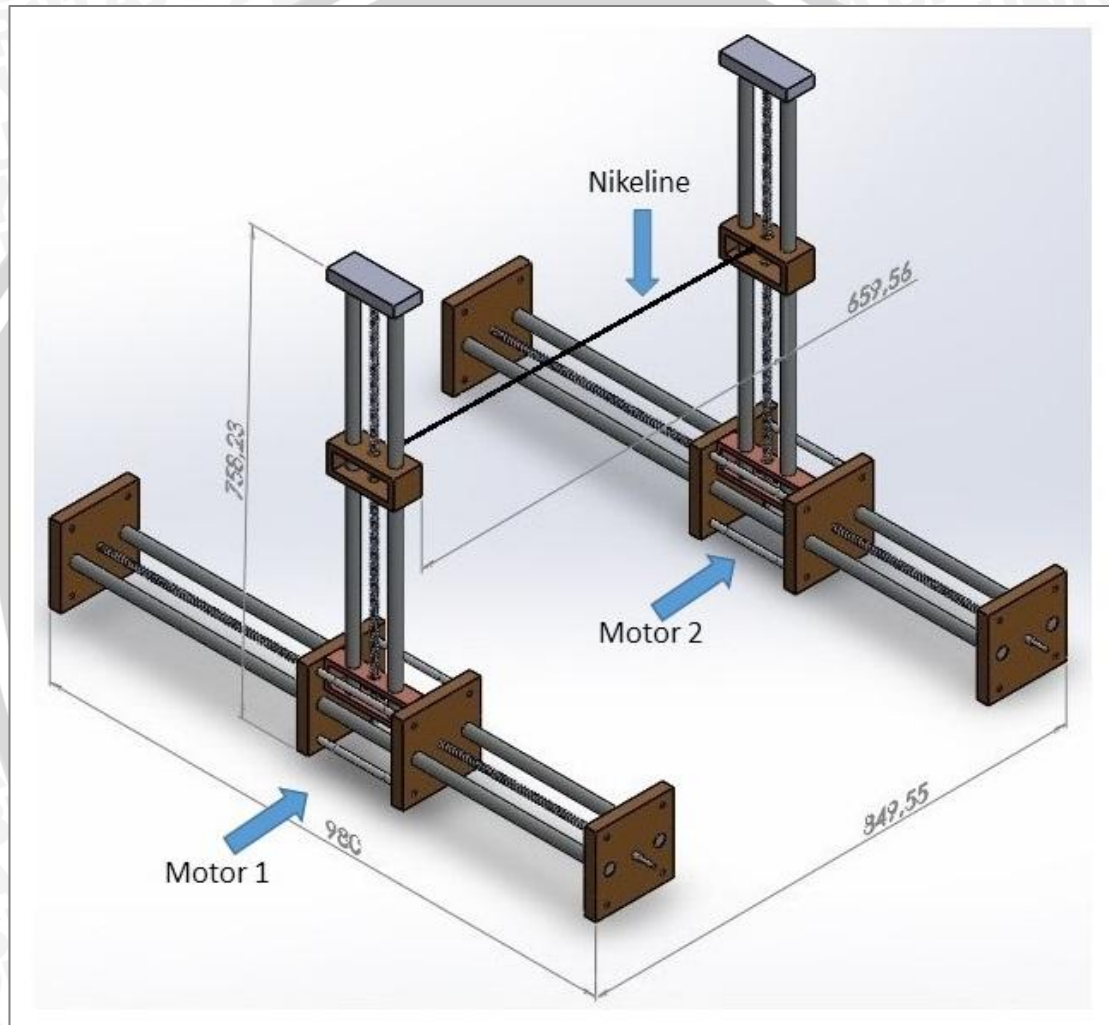
Sistem pemotongan *styrofoam* menggunakan kawat pemanas nikeline yang dibentangkan dan ditahan dikedua ujungnya dengan sistem mekanik yang dapat bergerak naik atau turun. Untuk menggerakkan dikedua sisinya digunakan dua buah motor listrik yang diatur kestabilan kecepatan gerakannya secara bersamaan. Kecepatan gerak kedua motor harus distabilkan agar dapat tahan terhadap gangguan-gangguan selama proses pemotongan. Selain itu agar diperoleh gerakan yang sama sampai akhir pemotongan. Spesifikasi mekanik pemotongan adalah sebagai berikut:

1. Panjang 980 mm
2. Lebar 350,0 mm
3. Tinggi 758,0 mm
4. Panjang maksimal nikeline 660,0 mm

Sistem mekanik *hot-wire cutting* secara lengkap diperlihatkan dalam Gambar 4.1.

Kawat nikeline sebagai pemanas untuk pemotongan, kedua ujungnya ditahan pada

mekanik yang dapat bergerak dengan bantuan aktuator motor listrik. Motor listrik yang digunakan memiliki spesifikasi tegangan 24 Volt. Selanjutnya kecepatan gerak kedua motor diatur dengan kontroler PID agar memiliki kecepatan gerak yang sama. Umpan balik yang digunakan berupa sensor *optocoupler* yang diletakkan pada salah satu sisi motor listrik.



**Gambar 4. 1** Sistem mekanik *hot-wire cutting*

#### 4.2 Spesifikasi Sistem Kontroler

Perancangan alat ini dilakukan bertahap dalam bentuk diagram blok sehingga memudahkan dalam analisis pada setiap bloknya maupun secara keseluruhan sistem. Perancangan ini terdiri atas:

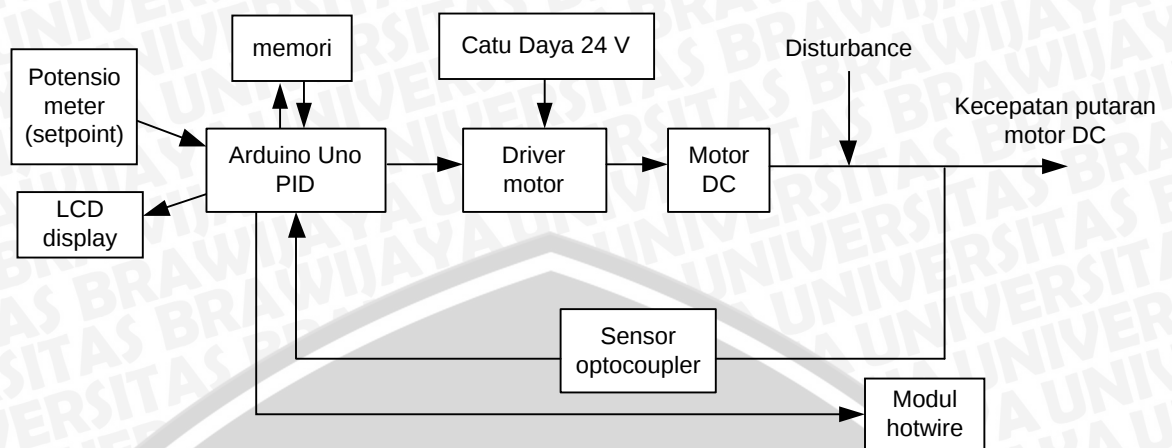
1. Perancangan perangkat keras, Berdasar pada blok diagram perancangan alat yang telah disusun, perancangan perangkat keras meliputi perancangan rangkaian sensor kecepatan (*optocoupler*), rangkaian *driver* motor, motor DC dan modul mikrokontroler.
2. Perancangan perangkat lunak (perancangan algoritma PID pada *software* Arduino Uno ERW 1.0.5).

Spesifikasi alat yang di rancang adalah sebagai berikut:

1. Arduino Uno R3 dan software Arduino ERW 1.0.5 untuk pengembangan kontroler Proporsional Integral Derferensial (PID).
2. Sensor kecepatan berupa menggunakan sensor *optocoupler* berfungsi sebagai *feedback* kecepatan putar motor DC.
3. Driver motor DC, untuk mengatur tegangan motor DC digunakan metode PWM dengan menggunakan MOSFET IRF640.
4. Motor DC yang digunakan adalah motor DC 24 Volt, 3500 rpm
5. *Hotwire* merupakan kawat panas untuk memotong *styrofoam*.

#### 4.3 Diagram Blok Sistem

Dalam skripsi ini dibuat diagram blok agar dalam pengerjaan dapat dilakukan sesuai dengan rancangan sistem. Adapun diagram blok tersebut dapat dilihat pada Gambar 4.2:



**Gambar 4. 2** Diagram Blok Sistem

Keterangan dari diagram blok dalam Gambar.4.2:

1. Masukan/*setpoint* berupa kecepatan putaran diberikan melalui *program* pada Arduino Uno.
2. Data masukan *setpoint* tersebut digunakan oleh mikrokontroler untuk mengatur kecepatan motor DC sebagai *setpoint* untuk memperoleh *error*.
3. Kemudian input diolah dan menghasilkan sinyal kontrol berupa PWM yang akan mejadi masukan untuk menggerakan motor DC.
4. Mikrokontroler mengaktifkan modul *hot wire* dan menggerakan motor secara *linear*.
5. Kecepatan putaran motor DC akan dihitung oleh mikrokontroler berdasarkan lebar pulsa yang dihasilkan oleh sensor *optocoupler*.
6. Mikrokontroler menghitung besarnya *error* berdasarkan perbedaan antara besarnya *setpoint* dengan besarnya kecepatan sesuai hasil pada poin 4.
7. Berdasarkan algoritma kontroler PID, keluaran dari PID dihitung berdasarkan besarnya *error* yang telah dihitung dalam poin 5. Selanjutnya hasil perhitungannya dikeluarkan ke PWM mikrokontroler untuk mengatur besarnya tegangan motor DC.

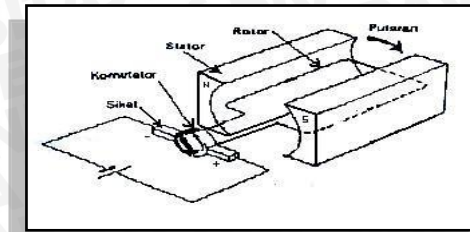
8. Pada saat sistem bekerja, operator bisa mengaktifkan atau menonaktifkan kontroler sesuai dengan kondisi yang diinginkan. Pada saat kontroler aktif sistem bekerja dan modul *hotwire* diaktifkan. Sebaliknya, pada saat kontroler nonaktif modul dinonaktifkan.

#### 4.4 Perancangan Perangkat Keras

##### 4.4.1 Motor DC

Motor DC merupakan motor arus searah yaitu suatu mesin yang berfungsi mengubah tenaga listrik arus searah menjadi tenaga gerak atau tenaga mekanik, di mana gerak tersebut berupa putaran dari pada rotor.

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 4.3 di bawah ini:

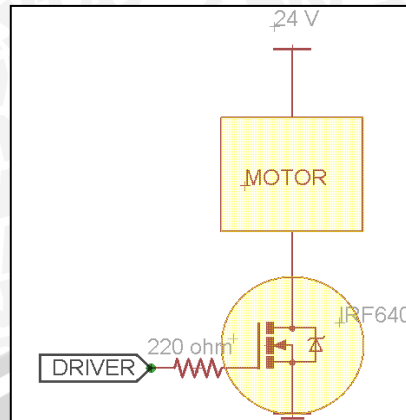


**Gambar 4. 3** Motor DC

Sumber : Kismet F [1994:98]

#### **4.4.2 Driver Motor DC**

*Driver* motor digunakan sebagai penghubung antar mikrokontroler ke motor DC. Digunakan driver motor karena arus yang keluar dari mikrokontroler tidak mampu mencukupi kebutuhan dari motor DC. Rangkaian *driver* motor dengan utama transistor, mosfet dan relay. Relay digunakan untuk membalik polaritas motor yang diaktifkan oleh transistor. Sedangkan mosfet digunakan untuk mengatur kecepatan motor DC. Nilai yang diberikan pada input motor digunakan untuk menentukan arah putaran motor. Nilai yang diberikan pada input PWM digunakan untuk mengatur kecepatan motor. Rangkaian driver motor DC dalam Gambar 4.4.



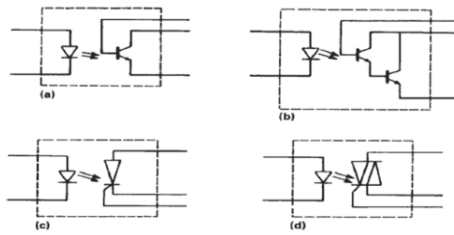
**Gambar 4. 4** Rangkaian *driver* motor DC

Sumber : Perancangan

#### 4.4.3 Sensor *Optocoupler*

Komponen yang digunakan untuk mengisolasi komponen elektronika daya dengan rangkaian berdaya rendah yang sebagai rangkaian pengontrol salahsatunya adalah *optocoupler*. Sering kali ditemukan rangkaian elektronika daya orde megawatt dikontrol hanya dengan rangkaian beberapa watt. *Optocoupler* berfungsi sebagai pengisolasi sinyal listrik antara rangkaian sinyal masukan dan keluaran (Mazda, 1997: 82). *Optocoupler* merupakan alat yang digunakan untuk mengkopel suatu sumber dengan detektor menggunakan perantara cahaya. Oleh karena itu piranti ini disebut dengan *optoisolator* / *optocoupler*. *Optocoupler* terbuat dari kombinasi dari sumber cahaya dan pendeteksi cahaya yang tergabung dalam satu kemasan. *Light emitting diode* atau *LED* seringkali digunakan sebagai sumber cahaya, dan berbagai komponen semikonduktor sebagai detektor cahaya. Kaca atau plastik digunakan sebagai pemisah antara sumber cahaya dan detektor, sehingga memungkinkan untuk ditempatkan pada tempat yang berdekatan. Pada prinsipnya, sinyal listrik dalam bentuk arus pada masukan diubah menjadi sinyal

optik dengan menggunakan sumber cahaya LED. Sinyal optik tersebut akan diterima oleh detektor untuk diubah kembali menjadi sinyal listrik. Umumnya *optocoupler* dipakai untuk mengisolasi sinyal listrik yang ada pada rangkaian masukan dan keluaran sehingga dapat digunakan transmisi sinyal antar rangkaian. *Optocoupler* memiliki beberapa *parameter* antara lain: tingkat isolasi antarasumber cahaya dan detektor, rasio transfer arus masukan-luaran, dan kecepatan operasi *optocoupler*. Rasio *transfer* arus merupakan rasio antara arus keluaran dan arus masukan dari sumber potensial, ketika *optocoupler* bekerja. Berbagai jenis dari detektor yang digunakan pada *optocoupler* ditunjukkan pada Gambar 6.10 foto transistor-*coupler* dikenal dalam kategori harga rendah dengan kecepatan operasi kerja 100-500 kHz dan minimum rasio *transfer* arus antara 20% dan 300%. Foto darlington memiliki rasio *transfer* arus antara 100% dan 1000% namun memiliki tingkat keakuratan yang rendah dikarenakan adanya tingkat penguatan pada hubungan darlington dengan kecepatan operasi termasuk rendah, dengan nilai antara 20 kHz dan 10 kHz. Foto thyristor dan foto triac digunakan sebagai detector untuk aplikasi dengan arus yang besar. Namun *parameter* yang perlu diperhatikan lebih lanjut adalah arus yang mengalir melalui *LED* yang nantinya akan memicu thyristor atau TRIAC. Efisiensi hantaran antara *LED* dengan foto transistor yang rendah sehingga perlu untuk mendesain thyristor atau TRIAC yang memiliki tingkat sensitivitas yang tinggi. Biasanya proses desain memerlukan perhatian lebih agar tidak menurunkan *parameter* yang lain seperti kemampuan hantaran arus. Beberapa *parameter* umum dari foto transistor dan foto triac antara lain : arus pemicuan 10-130 mA, mampu menyalurkan 100 sampai 300 mA, dan Waktu kerja 1-10 ms.



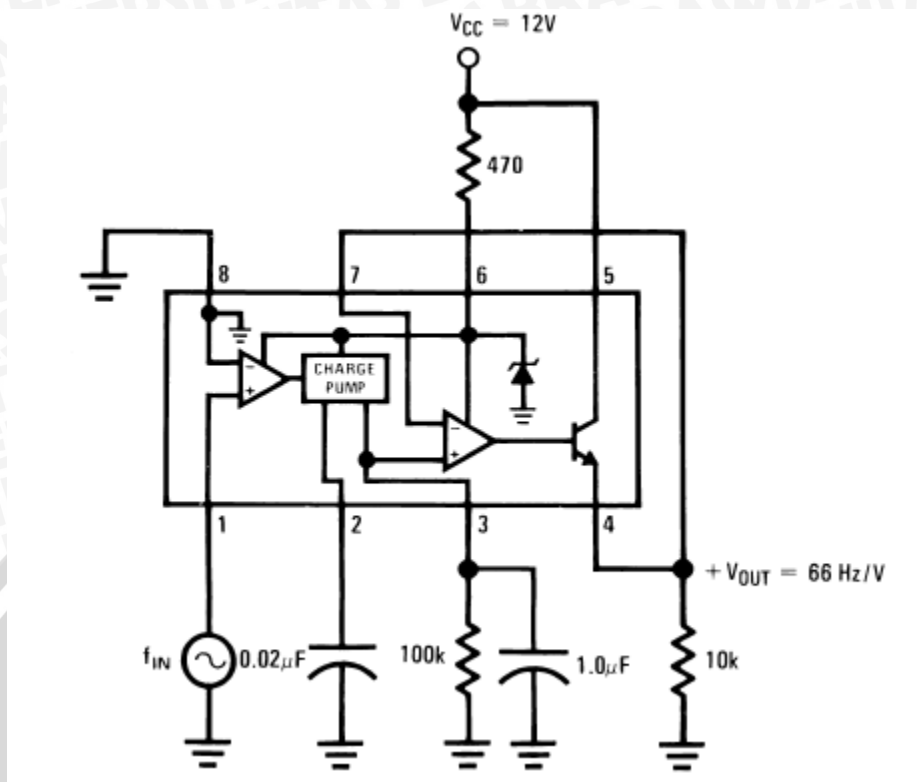
**Gambar 4. 5** Beberapa contoh detektor optik aplikasi elektronika daya.

a) Transistor, b) Darlington, c) Thyristor, d) Triac

Sumber : Mazda (1997:83)

#### 4.4.4 Rangkaian *Frequency to Voltage*

*Frequency to Voltage* digunakan untuk mengubah sinyal digital dari keluaran sensor *optocoupler* menjadi sinyal analog agar dapat dimasukkan pada board Arduino Uno. Rangkaian konversi frekuensi ke analog menggunakan *integrated circuit* (IC) LM2917 yang memiliki tegangan kerja +12 volt DC hingga +24 volt DC. Skema rangkainnya dapat ditunjukkan pada Gambar 4.6.



Gambar 4. 6 Skematik Rangkaian Frequency to Voltage

Sumber: Datasheet LM2917

Berdasarkan pada *datasheet* dengan rangkaian seperti pada gambar 4.6, maka tegangan keluaran dapat dihitung dengan menggunakan persamaan:

$$V_{out} = \frac{f_{in}}{66} \dots\dots\dots(4-1)$$

Atau secara umum perhitungan yang digunakan pada rangkaian yang menggunakan IC LM2907/2917 dapat dijelaskan pada persamaan berikut.

$$V_{out} = f_{in} \times V_{cc} \times R_1 \times C_1 \dots\dots\dots(4-2)$$

dimana :

$f_{in}$  = Frekuensi sinyal input (Hz)

$V_{cc}$  = Tegangan sumber yang digunakan (volt)

$R_1$  = Resistor pada pin 3 IC LM2917 (Ohm)

$C_1$  = Kapasitor pada pin 2 IC LM2917 (Farad)

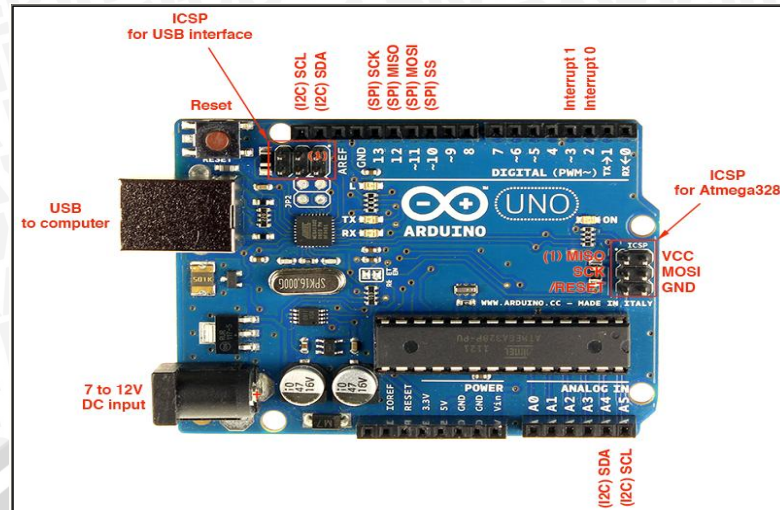
#### 4.4.5 Penentuan Nilai Penguatan Kontroler

Untuk menentukan penguatan kontroler ( $K_p$ ,  $K_i$ ,  $K_d$ ) digunakan metode *hand tuning* (*hand eksperimen*) untuk mendapatkan hasil respon sesuai dengan yang diinginkan. *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 ( $K_d$ ). Proses *tuning* parameter PID ini dilakukan dengan cara mengatur nilai  $K_p$  hingga didapatkan respon sistem yang mendekati *setpoint* 3500 RPM.

#### 4.4.6 Arduino Uno R3

Arduino Uno adalah sebuah *board* mikrokontroler berbasis ATmega328. Arduino Uno mempunyai 14 pin *digital input/output* (6 di antaranya dapat digunakan sebagai *output* PWM), 6 input analog, sebuah osilator Kristal 16 MHz, sebuah koneksi USB, sebuah *power jack*, sebuah ICSP *header*, dan sebuah tombol *reset*. Arduino Uno memuat semua yang dibutuhkan untuk menunjang mikrokontroler, mudah menghubungkannya ke sebuah komputer dengan sebuah kabel USB atau mensuplainya dengan sebuah *adaptor* AC ke DC atau menggunakan baterai untuk memulainya.

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.



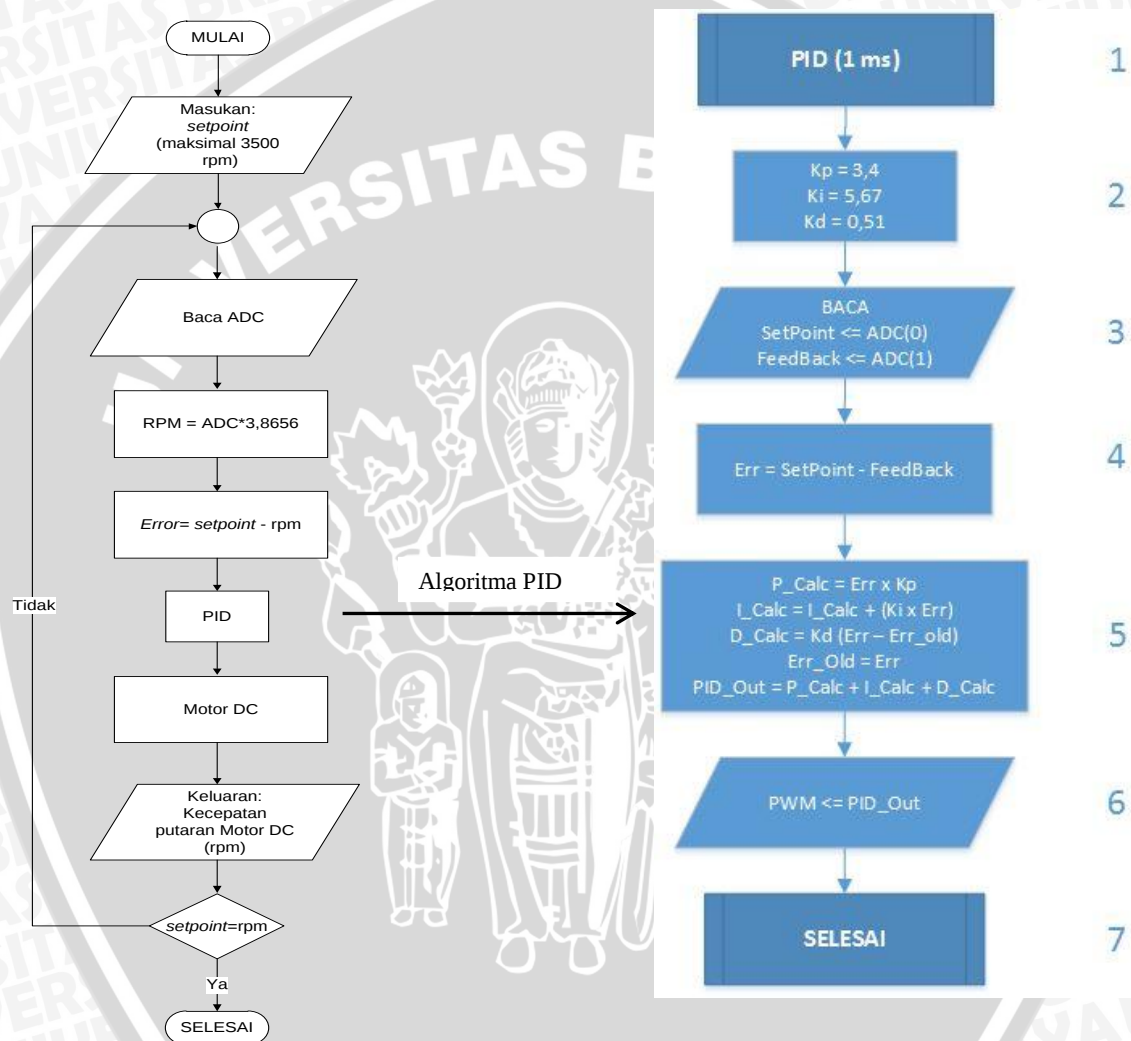
**Gambar 4. 7** Board Arduino Uno R3

Sumber : [electroschematics.com](http://electroschematics.com)



#### 4.5 Perancangan Perangkat Lunak

Perancangan perangkat lunak pada pengendalian ini menggunakan bahasa pemrograman C++ dengan menggunakan *software* Arduino ERW 1.0.5 Flowchart perancangan perangkat lunak dapat dilihat pada Gambar 4.8.



Gambar 4. 8 Flowchart Perangkat Lunak

## BAB V

### PENGUJIAN DAN ANALISIS SISTEM

Pengujian alat bertujuan untuk mengetahui sejauh mana keberhasilan perancangan sistem. Pengujian alat dilakukan pada bagian perbagian dari sistem dan pengujian cara kerja secara keseluruhan. Seperti yang diperlihatkan dalam perancangan sistem secara diagram blok pada Gambar 4.2, pengujian alat tahap demi tahap adalah:

1. Pengujian sensor kecepatan putar motor DC.
2. Pengujian respon sistem secara *open loop*.
3. Penentuan konstanta kontroler  $K_p$ ,  $K_d$  dan  $K_i$ .
4. Pengujian respon sistem secara *closed loop*.

#### 5.1 Pengujian sensor kecepatan putar motor DC

##### 5.1.1 Tujuan

Pengujian ini bertujuan untuk mengetahui hubungan antara kecepatan putar dengan tegangan keluaran *Frequency to Voltage LM2917*.

##### 5.1.2 Peralatan yang digunakan

1. RPM Meter optik
2. Voltmeter Digital
3. Power Supply DC 24Volt

##### 5.1.3 Langkah Pengujian

1. Menyusun rangkaian *Frequency to Voltage LM2917*.
2. Memberikan catu daya motor DC dimulai dengan 1 Volt
3. Mengukur RPM dengan menggunakan RPM Meter

4. Mengukur tegangan keluaran rangkaian *Frequency to Voltage LM291*.
5. Mengulangi langkah 2 dengan penambahan tegangan 0,5V sampai dengan 24Volt.

#### 5.1.4 Hasil Pengujian dan analisis

Hasil pengujian diperlihatkan dalam Tabel 5.1

**Tabel 5.1** Hasil Pengujian sensor kecepatan putar motor DC

| No | Tegangan Motor DC (Volt) | RPM Meter | Tegangan Keluaran (Volt) |
|----|--------------------------|-----------|--------------------------|
| 1  | 1                        | 174       | 0,04                     |
| 2  | 1,5                      | 261       | 0,066                    |
| 3  | 2                        | 349       | 0,088                    |
| 4  | 2,5                      | 435       | 0,11                     |
| 5  | 3                        | 520       | 0,131                    |
| 6  | 3,5                      | 611       | 0,154                    |
| 7  | 4                        | 700       | 0,177                    |
| 8  | 4,5                      | 786       | 0,198                    |
| 9  | 5                        | 873       | 0,22                     |
| 10 | 5,5                      | 960       | 0,242                    |
| 11 | 6                        | 1049      | 0,265                    |
| 12 | 6,5                      | 1137      | 0,287                    |
| 13 | 7                        | 1224      | 0,31                     |
| 14 | 7,5                      | 13210     | 0,336                    |
| 15 | 8                        | 1400      | 0,354                    |
| 16 | 8,5                      | 1485      | 0,374                    |
| 17 | 9                        | 1576      | 0,398                    |
| 18 | 9,5                      | 1666      | 0,421                    |
| 19 | 10                       | 1755      | 0,443                    |
| 20 | 10,5                     | 1830      | 0,462                    |
| 21 | 11                       | 1923      | 0,486                    |
| 22 | 11,5                     | 2011      | 0,508                    |
| 23 | 12                       | 2110      | 0,533                    |
| 24 | 12,5                     | 2190      | 0,553                    |
| 25 | 13                       | 2273      | 0,574                    |
| 26 | 13,5                     | 2370      | 0,598                    |
| 27 | 14                       | 2455      | 0,62                     |
| 28 | 14,5                     | 2540      | 0,641                    |
| 29 | 15                       | 2630      | 0,664                    |
| 30 | 15,5                     | 2715      | 0,686                    |
| 31 | 16                       | 2812      | 0,71                     |

| No | Tegangan Motor DC (Volt) | RPM Meter | Tegangan Keluaran (Volt) |
|----|--------------------------|-----------|--------------------------|
| 32 | 16,5                     | 2887      | 0,729                    |
| 33 | 17                       | 2980      | 0,753                    |
| 34 | 17,5                     | 3065      | 0,774                    |
| 35 | 18                       | 3155      | 0,797                    |
| 36 | 18,5                     | 3238      | 0,817                    |
| 37 | 19                       | 3324      | 0,839                    |
| 38 | 19,5                     | 3413      | 0,862                    |
| 39 | 20                       | 3525      | 0,89                     |
| 40 | 20,5                     | 3590      | 0,907                    |
| 41 | 21                       | 3680      | 0,93                     |
| 42 | 21,5                     | 3770      | 0,952                    |
| 43 | 22                       | 3855      | 0,973                    |
| 44 | 22,5                     | 3940      | 0,995                    |
| 45 | 23                       | 4030      | 1,02                     |
| 46 | 23,5                     | 4120      | 1,04                     |
| 47 | 24                       | 4210      | 1,06                     |

Tegangan keluaran akan dibaca oleh mikrokontroler sebagai sensor RPM motor DC menggunakan internal ADC 10bit dengan referensi ADC 1volt sebagai maksimal tegangannya. Berdasarkan rumus  $V_{out} = \frac{f_{in}}{66}$  dapat dikembalikan sebagai RPM yang sebenarnya yaitu:

$$RPM = V_{out} \times 66 \times 60 \dots\dots\dots(5.1)$$

Sebagai contoh data nomor 25, Vout (tegangan keluaran) = 0,574. RPM yang terbaca oleh mikrokontroler seharusnya:

$$RPM = 0,574 \times 66 \times 60 = 2273,04 \text{ RPM}$$

Sedangkan hasil ADC mikrokontroler akan diperoleh sebesar:

$$ADC = \frac{0,574}{1} \times 1024 = 588$$

Dengan cara yang sama RPM sensor dengan hasil ADC diperlihatkan dalam Tabel

5.2.

**Tabel 5.2** Hasil konversi RPM

| No | Tegangan Keluaran (Volt) | RPM Sensor | ADC |
|----|--------------------------|------------|-----|
| 1  | 0,04                     | 158        | 41  |
| 2  | 0,066                    | 261        | 68  |
| 3  | 0,088                    | 348        | 90  |
| 4  | 0,11                     | 436        | 113 |
| 5  | 0,131                    | 519        | 134 |
| 6  | 0,154                    | 610        | 158 |
| 7  | 0,177                    | 701        | 181 |
| 8  | 0,198                    | 784        | 203 |
| 9  | 0,22                     | 871        | 225 |
| 10 | 0,242                    | 958        | 248 |
| 11 | 0,265                    | 1049       | 271 |
| 12 | 0,287                    | 1137       | 294 |
| 13 | 0,31                     | 1228       | 317 |
| 14 | 0,336                    | 1331       | 344 |
| 15 | 0,354                    | 1402       | 362 |
| 16 | 0,374                    | 1481       | 383 |
| 17 | 0,398                    | 1576       | 408 |
| 18 | 0,421                    | 1667       | 431 |
| 19 | 0,443                    | 1754       | 454 |
| 20 | 0,462                    | 1830       | 473 |
| 21 | 0,486                    | 1925       | 498 |
| 22 | 0,508                    | 2012       | 520 |
| 23 | 0,533                    | 2111       | 546 |
| 24 | 0,553                    | 2190       | 566 |
| 25 | 0,574                    | 2273       | 588 |
| 26 | 0,598                    | 2368       | 612 |
| 27 | 0,62                     | 2455       | 635 |
| 28 | 0,641                    | 2538       | 656 |
| 29 | 0,664                    | 2629       | 680 |
| 30 | 0,686                    | 2717       | 702 |
| 31 | 0,71                     | 2812       | 727 |
| 32 | 0,729                    | 2887       | 746 |
| 33 | 0,753                    | 2982       | 771 |
| 34 | 0,774                    | 3065       | 793 |
| 35 | 0,797                    | 3156       | 816 |
| 36 | 0,817                    | 3235       | 837 |
| 37 | 0,839                    | 3322       | 859 |
| 38 | 0,862                    | 3414       | 883 |
| 39 | 0,89                     | 3524       | 911 |

| No | Tegangan Keluaran (Volt) | RPM Sensor | ADC  |
|----|--------------------------|------------|------|
| 40 | 0,907                    | 3592       | 929  |
| 41 | 0,93                     | 3683       | 952  |
| 42 | 0,952                    | 3770       | 975  |
| 43 | 0,973                    | 3853       | 996  |
| 44 | 0,995                    | 3940       | 1019 |
| 45 | 1,02                     | 4039       | 1023 |
| 46 | 1,04                     | 4118       | 1023 |
| 47 | 1,06                     | 4198       | 1023 |

Data ke 45 sampai 47 memiliki nilai yang sama dengan demikian nilai RPM maksimal yang dapat disensor adalah 3940 RPM, karena tegangan melebihi 1 Volt. Data yang terbaca mikrokontroler melalui ADC perlu adanya faktor pengali agar sesuai dengan RPM yang terukur. Sebagai contoh data ke 25 besarnya RPM adalah 2273, sedangkan hasil konversi ADC sebesar 588. Dengan demikian faktor pengalinya adalah  $2273/588=3,8656$ .

## 5.2 Pengujian respon sistem secara *open loop*

### 5.2.1 Tujuan

Pengujian ini bertujuan untuk mengetahui respon sistem secara *open loop* berguna sebagai dasar untuk menentukan konstanta kontroler  $K_p$ ,  $K_d$  dan  $K_i$ .

### 5.2.2 Peralatan yang digunakan

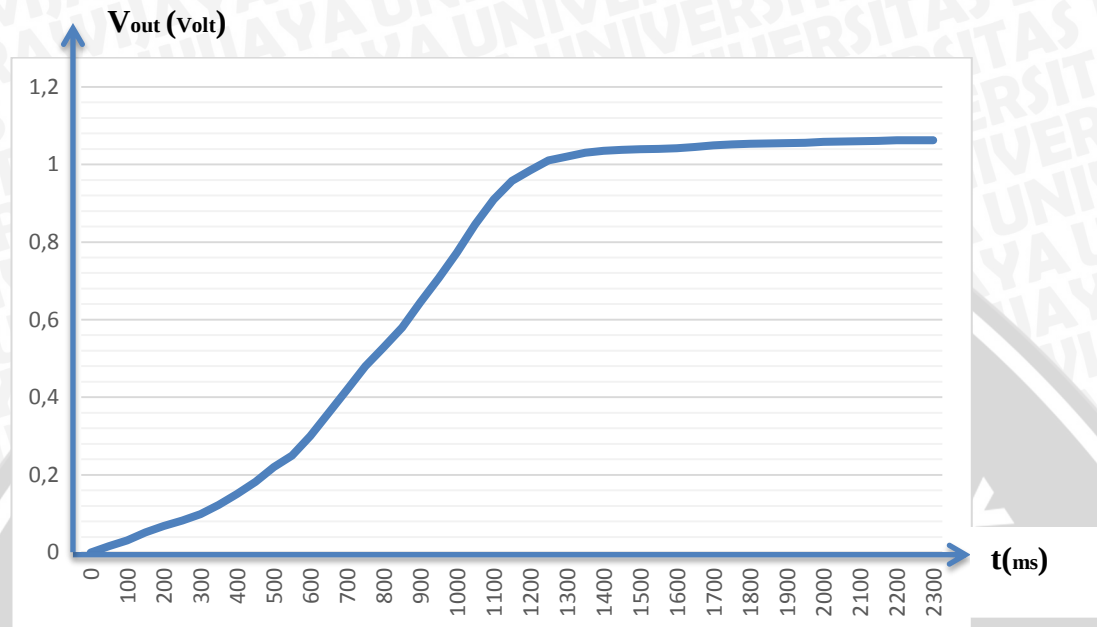
1. *Oscilloscope Digital*
2. Motor DC 24 Volt dan rangkaian sensor
3. *Power Supply* DC 12Volt

### 5.2.3 Langkah Pengujian

1. Merangkai peralatan keluaran sensor dihubungkan ke *Oscillocope*
2. Mengeset *Oscilloscope* pada 1Volt/Div
3. *Oscilloscope* dioperasikan sebagai data logger
4. Memberikan catu 24Volt pada motor DC
5. Merekam keluaran tegangan sensor

#### 5.2.4 Hasil Pengujian dan analisis

Hasil pengujian respon *open loop* motor DC diperlihatkan dalam Gambar 5.1



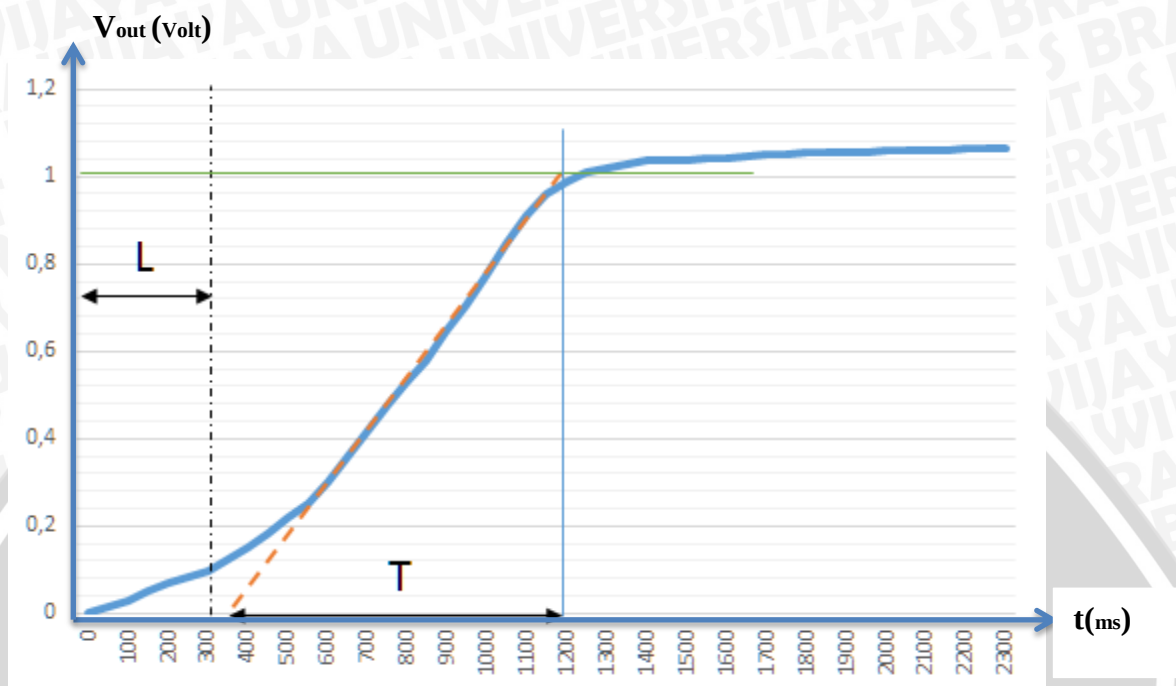
Gambar 5. 1 Hasil pengujian respon *open loop* motor DC

#### 5.3 Analisis penentuan konstanta kontroler $K_p$ , $K_d$ dan $K_i$ .

Dari hasil pengujian yang diperlihatkan dalam Gambar 5.1 berupa kurva S akan digunakan menentukan parameter *tuning* PID dengan metode 1 Ziegler-Nichols seperti yang diperlihatkan dalam Gambar 5.2. Adapun langkah-langkah yang dilakukan sebagai berikut:

1. Menarik garis tangent pada titik infleksi grafik karakteristik *open loop* seperti yang diperlihatkan dalam Gambar 5.2
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. Nilai L dan T digunakan untuk menentukan nilai Kp Ti dan Td sesuai dengan Tabel 5.3



Gambar 5. 2 Metode 1 Ziegler-Nichols (hasil pengujian)

Tabel 5.3 Aturan Metode 1 Ziegler-Nichols (Ogata K., 1997)

| Tipe Kontrol | Kp                   | $\tau_i$        | $\tau_d$ |
|--------------|----------------------|-----------------|----------|
| P            | $\frac{\tau}{L}$     | $\infty$        | 0        |
| PI           | $0,9 \frac{\tau}{L}$ | $\frac{L}{0,3}$ | 0        |
| PID          | $1,2 \frac{\tau}{L}$ | $2L$            | $0,5L$   |

Dalam Gambar 5.2, besarnya *steady state* ketika keluaran sensor adalah maksimal yaitu sebesar 1Volt yang ekivalen dengan  $1 \times 66 \times 60 = 3960$  RPM. Dari Gambar 5.2 diperoleh besarnya  $L = 300$ mS dan  $T = 1200 - 350 = 850$ mS.

Dengan demikian parameter kontroler diperoleh sebagai berikut:

$$K_p = 1,2 \frac{\tau}{L} = 1,2 \times \frac{0,850S}{0,300S} = 3,4$$

$$\tau_i = 2L = 2 \times 0,300S = 0,6$$

$$\tau_d = 0,5L = 0,5 \times 0,300S = 0,15$$

Selanjutnya akan diperoleh  $K_i$  dan  $K_d$  sebagai berikut:

$$K_i = \frac{Kp}{\tau_i} = \frac{3,4}{0,6} = 5,67$$

$$K_d = Kp \times \tau_d = 3,4 \times 0,15 = 0,51$$

## 5.4 Pengujian respon sistem secara *closed loop*

### 5.4.1 Tujuan

Pengujian ini bertujuan untuk mengetahui respon sistem secara *closed loop* dengan memberikan gangguan sementara.

### 5.4.2 Peralatan yang digunakan

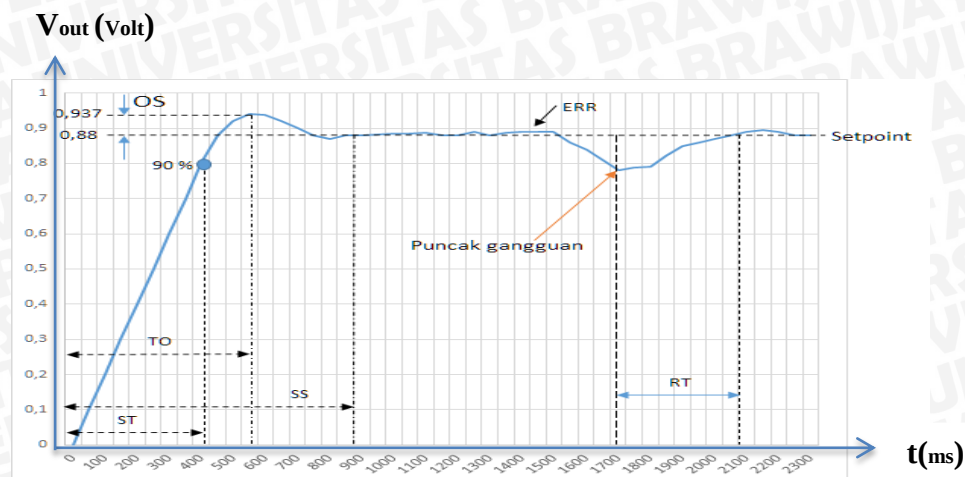
1. *Oscilloscope Digital*
2. Motor DC 24 Volt dan rangkaian sensor
3. *Power Supply* DC 12Volt
4. Balok *Styrofoam*

### 5.4.3 Langkah Pengujian

1. Merangkai peralatan keluaran sensor dihubungkan ke *Oscilloscope*
2. Mengeset *Oscilloscope* pada 1Volt/Div
3. *Oscilloscope* dioperasikan sebagai datalogger
4. Memberikan *setpoint* pada 3500RPM
5. Merekam keluaran tegangan sensor
6. Memberikan gangguan.

### 5.4.4 Hasil Pengujian dan analisis

Hasil pengujian respon *closed loop* secara grafis tanpa gangguan dan gangguan diperlihatkan dalam Gambar 5.3.



**Gambar 5. 3** Hasil pengujian respon sistem secara *closed loop*

Dari hasil pengujian secara *closed loop* seperti yang diperlihatkan dalam Gambar 5.3 dapat diperoleh parameter unjuk kerja sistem sebagai berikut:

1. *Time overshoot* (TO) terjadi pada sekitar 0,6 detik
2. *Settling Time* (ST) terjadi pada sekitar 0,45 detik
3. *Steady State* terjadi pada sekitar 0,9 detik
4. Besarnya *Overshoot* (OS) sebesar  $\frac{0,937-0,88}{0,88} \times 100\% = 5,7\%$
5. *Error steady state* adalah nilai kesalahan saat respon telah mencapai keadaan tunak / steady. Presentase kesalahan dapat dicari dengan persamaan dibawah:

$$\%E = \frac{1}{n} \sum_{i=1}^n \frac{|PV-SP|}{SP} \times 100\%$$

$$\%E = \frac{1}{12} \times 0,063 \times 100\%$$

$$\%E = 0,525\%$$

Dimana n adalah jumlah data ketika telah mencapai *setpoint*, PV adalah *present value*, dan SP adalah *setpoint*.

6. *Recovery Time* (RT) setelah adanya gangguan sebesar 2100-1700 = 400mS.

## BAB VI

### KESIMPULAN DAN SARAN

#### 6.1 Kesimpulan

Dari perancangan kemudian dilakukan pengujian dari keseluruhan sistem, selanjutnya dapat disimpulkan sebagai berikut:

1. Alat yang dirancang sudah sesuai dengan yang diharapkan, dimana kecepatan gerak pemotongan dikedua sisi sudah sesuai yang diinginkan.
2. Dengan menggunakan metode 1 *Ziegler-Nichols* untuk menentukan nilai parameter kontroler PID, maka didapatkan nilai  $K_p = 3,4$ ,  $K_i = 5,67$  dan  $K_d = 0,51$
3. Setelah diimplementasikan, sistem dapat berjalan sesuai dengan yang diharapkan. Dengan rata-rata %*E* kurang dari 2%, dan maksimum *overshoot* 5,7%

#### 6.2 Saran

Sebagai pengembangan selanjutnya terdapat beberapa saran-saran sebagai berikut:

1. Sistem dapat dikembangkan untuk membuat pola-pola yang lain seperti segitiga, lingkaran, kotak, dan sebagainya.
2. Dikembangkan dengan komputer agar bisa sebagai CNC (*Computer Numerically Control*).

## DAFTAR PUSTAKA

- Astrom, K.J, & Hagglund, Tore. 1995. *PID Controllers: Theory, Design and Tuning*. Research Triangle Park: Instrument Society of America.
- Atmel Corporation. 2019. *ATMEGA 328 Series*.
- Gunterus, Frans. 1994. *Falsafah Dasar : Sistem Pengendalian Proses*. Jakarta: Elex Media Komputindo.
- Ogata, Katsuhiko. 1997. *Teknik Kontrol Automatik*. Jakarta: Penerbit Erlangga.
- Mazda, Fraidoon. 1997. *Power Electronics Handbook 3rd Edition*. Oxford: Newnes.
- Smith, L. C. 1979. *Fundamentals of control theory*. Deskbook issue.
- Krisnanda, Ferditya. 2014. *Pengendalian Kecepatan Putaran Gas Engine Pada RC Airplane Menggunakan Kontroller Proporsional Integral Derferensial (PID) Berbasis Mikrokontroller ATmega 328*. Malang: Teknik Elektro Universitas Brawijaya.
- Albar, Doni Aditya. 2012. *Pengendalian Kecepataan Motor Servo Untuk Pemotongan Styrofoam Secara Radial Dengan Hotwire Berbasis Mikrokontroler ATmega 328*. Malang: Teknik Elektro Universitas Brawijaya.

## LAMPIRAN



## LAMPIRAN I FOTO ALAT



Foto Alat tampak depan



Foto Alat tampak samping

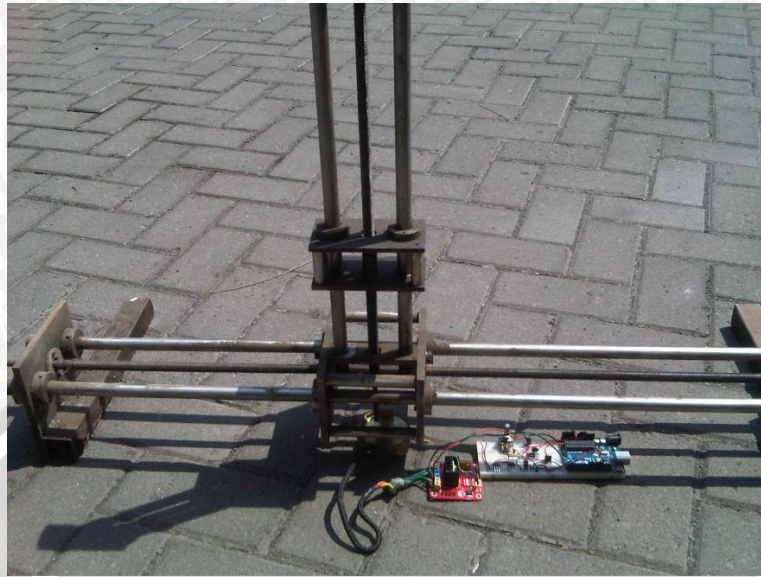
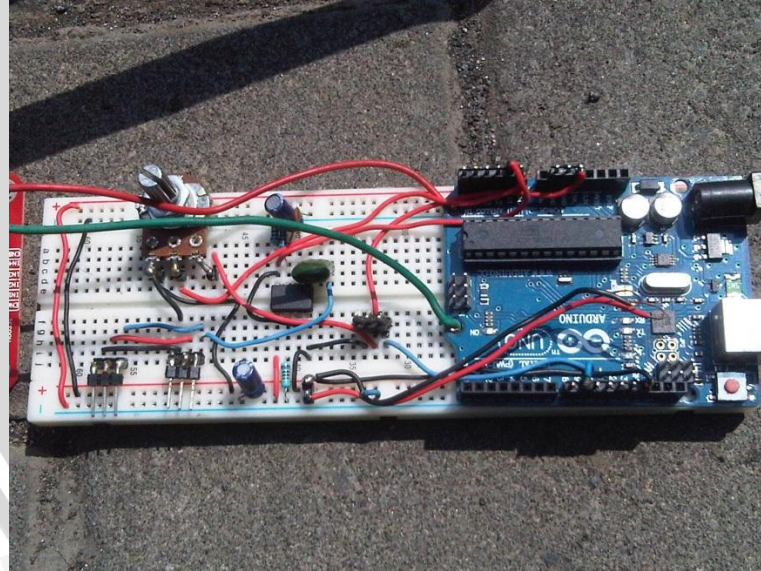


Foto Alat dan Rangkaian Elektrik



Arduino Mega dan Rancangan Rangkaian Elektrik

LAMPIRAN II*LISTING PROGRAM*

```
/******
```

```
* TUGAS AKHIR
```

```
* JUDUL:
```

```
* Pengendalian Kecepatan Putaran Motor DC untuk Pemotongan
```

```
* Styrofoam Secara Linear Menggunakan Kontroler Proporsional
```

```
* Integral Diferensial (PID) berbasis Mikrokontroler ATMEGA 328
```

```
* NAMA: Ventario Amanda
```

```
* NIM : 105060307111038
```

```
*****/
```

```
#include <PID_v1.h>
```

```
// Mendefinisikan kanal adc setpoint Input dan Output
```

```
double Setpoint, Input, Output;
```

```
double Kp, Ki, Kd
```

```
// setting parameter PID
```

```
Kp = 3,4;
```

```
Ki = 5,67;
```

```
Kd = 0,51;
```

```
PID myPID(&Input, &Output, &Setpoint,Kp,Ki,Kd, DIRECT);
```

```
void setup()
```

```
{
```

```
// inialisasi
// baca feedback kanal adc 0
Input = analogRead(0);
// baca setpoint kanal adc 1
Setpoint = analogRead(1);

//turn the PID on
myPID.SetMode(AUTOMATIC);
}

void loop()
{
    // baca feedback
    Input = analogRead(0);
    // baca setpoint
    Setpoint = analogRead(1);
    myPID.Compute();
    // keluarkan hasilnya
    analogWrite(3,Output);
}
```



```

/*****
*****
* Arduino PID Library - Version 1.1.1
* by Brett Beauregard <br3ttb@gmail.com> brettbeauregard.com
*
* This Library is licensed under a GPLv3 License
*****
*****/

#if ARDUINO >= 100
  #include "Arduino.h"
#else
  #include "WProgram.h"
#endif

#include <PID_v1.h>

/*Constructor
(...)*****
* The parameters specified here are those for for which we
can't set up
* reliable defaults, so we need to have the user set them.
*****
*****/
PID::PID(double* Input, double* Output, double* Setpoint,
        double Kp, double Ki, double Kd, int ControllerDirection)
{
    myOutput = Output;
    myInput = Input;
    mySetpoint = Setpoint;
    inAuto = false;

    PID::SetOutputLimits(0, 255);
    //default output limit corresponds to

    //the arduino pwm limits

    SampleTime = 100;
    //default Controller Sample Time is 0.1 seconds

    PID::SetControllerDirection(ControllerDirection);
    PID::SetTunings(Kp, Ki, Kd);

    lastTime = millis()-SampleTime;
}

```

```

/* Compute()
*****
****
*   This, as they say, is where the magic happens.  this
function should be called
*   every time "void loop()" executes.  the function will decide
for itself whether a new
*   pid Output needs to be computed.  returns true when the
output is computed,
*   false when nothing has been done.

*****
*****/
bool PID::Compute()
{
    if(!inAuto) return false;
    unsigned long now = millis();
    unsigned long timeChange = (now - lastTime);
    if(timeChange>=SampleTime)
    {
        /*Compute all the working error variables*/
        double input = *myInput;
        double error = *mySetpoint - input;
        ITerm+= (ki * error);
        if(ITerm > outMax) ITerm= outMax;
        else if(ITerm < outMin) ITerm= outMin;
        double dInput = (input - lastInput);

        /*Compute PID Output*/
        double output = kp * error + ITerm- kd * dInput;

        if(output > outMax) output = outMax;
        else if(output < outMin) output = outMin;
        *myOutput = output;

        /*Remember some variables for next time*/
        lastInput = input;
        lastTime = now;
        return true;
    }
    else return false;
}

/*
SetTunings(...)*****
*****
*   This function allows the controller's dynamic performance to
be adjusted.
*   it's called automatically from the constructor, but tunings
can also
*   be adjusted on the fly during normal operation

```

```

*****
*****/
void PID::SetTunings(double Kp, double Ki, double Kd)
{
    if (Kp<0 || Ki<0 || Kd<0) return;

    dispKp = Kp; dispKi = Ki; dispKd = Kd;

    double SampleTimeInSec = ((double)SampleTime)/1000;
    kp = Kp;
    ki = Ki * SampleTimeInSec;
    kd = Kd / SampleTimeInSec;

    if(controllerDirection ==REVERSE)
    {
        kp = (0 - kp);
        ki = (0 - ki);
        kd = (0 - kd);
    }
}

/* SetSampleTime(...)
*****
* sets the period, in Milliseconds, at which the calculation is
performed

*****
*****/
void PID::SetSampleTime(int NewSampleTime)
{
    if (NewSampleTime > 0)
    {
        double ratio = (double)NewSampleTime
                        / (double)SampleTime;
        ki *= ratio;
        kd /= ratio;
        SampleTime = (unsigned long)NewSampleTime;
    }
}

/*
SetOutputLimits(...)*****
*****
* This function will be used far more often than
SetInputLimits. while
* the input to the controller will generally be in the 0-1023
range (which is
* the default already,) the output will be a little different.
maybe they'll
* be doing a time window and will need 0-8000 or something. or
maybe they'll

```

```

* want to clamp it from 0-125.  who knows.  at any rate, that
can all be done
* here.

*****
*****/
void PID::SetOutputLimits(double Min, double Max)
{
    if(Min >= Max) return;
    outMin = Min;
    outMax = Max;

    if(inAuto)
    {
        if(*myOutput > outMax) *myOutput = outMax;
        else if(*myOutput < outMin) *myOutput = outMin;

        if(ITerm > outMax) ITerm= outMax;
        else if(ITerm < outMin) ITerm= outMin;
    }
}

/*
SetMode(...)*****
*****
* Allows the controller Mode to be set to manual (0) or
Automatic (non-zero)
* when the transition from manual to auto occurs, the controller
is
* automatically initialized

*****
*****/
void PID::SetMode(int Mode)
{
    bool newAuto = (Mode == AUTOMATIC);
    if(newAuto == !inAuto)
    { /*we just went from manual to auto*/
        PID::Initialize();
    }
    inAuto = newAuto;
}

/*
Initialize()*****
*****
* does all the things that need to happen to ensure a
bumpless transfer
* from manual to automatic mode.

*****
*****/

```

```

void PID::Initialize()
{
    ITerm = *myOutput;
    lastInput = *myInput;
    if(ITerm > outMax) ITerm = outMax;
    else if(ITerm < outMin) ITerm = outMin;
}

/*
SetControllerDirection(...)*****
*****
* The PID will either be connected to a DIRECT acting process
(+Output leads
* to +Input) or a REVERSE acting process(+Output leads to -
Input.) we need to
* know which one, because otherwise we may increase the output
when we should
* be decreasing. This is called from the constructor.

*****
*****/
void PID::SetControllerDirection(int Direction)
{
    if(inAuto && Direction !=controllerDirection)
    {
        kp = (0 - kp);
        ki = (0 - ki);
        kd = (0 - kd);
    }
    controllerDirection = Direction;
}

/* Status
Functions*****
****
* Just because you set the Kp=-1 doesn't mean it actually
happened. these
* functions query the internal state of the PID. they're here
for display
* purposes. this are the functions the PID Front-end uses for
example

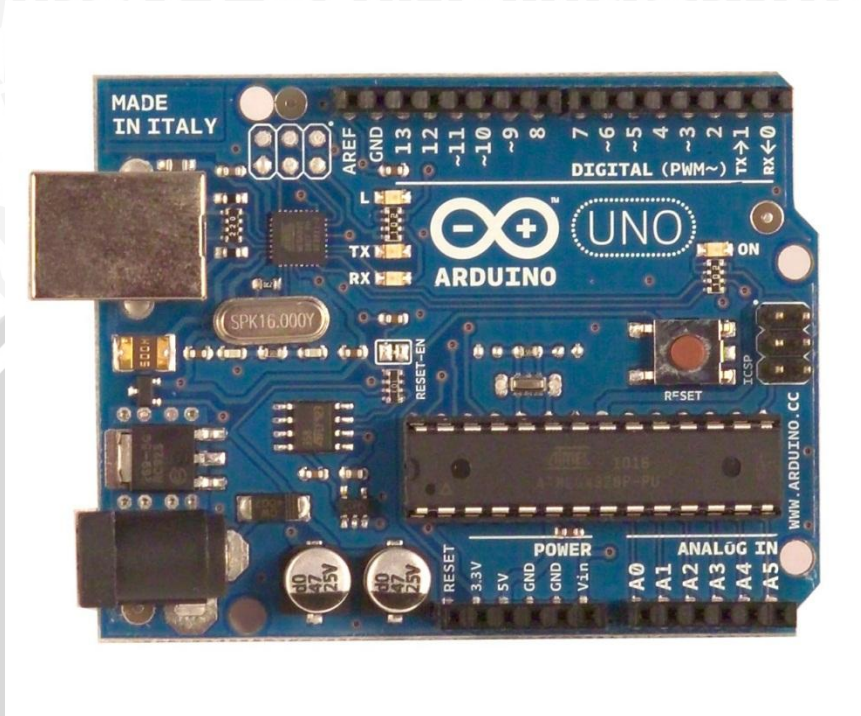
*****
*****/
double PID::GetKp(){ return dispKp; }
double PID::GetKi(){ return dispKi; }
double PID::GetKd(){ return dispKd; }
int PID::GetMode(){ return inAuto ? AUTOMATIC : MANUAL;}
int PID::GetDirection(){ return controllerDirection;}

```

**LAMPIRAN III**  
***DATASHEET***



# Arduino UNO



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

1.0 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](#).

## Technical Specification



|                             |                                          |
|-----------------------------|------------------------------------------|
| 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                                   |

### 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.



## 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 it is possible to change the upper end of their range using the AREF pin and the `analogReference()` function. Additionally, some pins have specialized functionality:

**I<sup>2</sup>C: 4 (SDA) and 5 (SCL).** Support I<sup>2</sup>C (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.

See also the [mapping between Arduino pins and Atmega328 ports](#).

## 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).

## 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

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.