



**UNIVERSITAS INDONESIA**

**RANCANG BANGUN HOTPLATE STIRRER MAGNETIK  
TERKENDALI TEMPERATUR DAN KECEPATAN  
PENGADUK**

**SKRIPSI**

**MUHAMAD AULIA RAHMAN**

**0906602124**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
PROGRAM STUDI FISIKA EKSTENSI  
DEPOK  
DESEMBER 2011**



**UNIVERSITAS INDONESIA**

**RANCANG BANGUN HOTPLATE STIRRER MAGNETIK  
TERKENDALI TEMPERATUR DAN KECEPATAN  
PENGADUK**

**SKRIPSI**

**Diajukan sebagai salah satu syarat untuk memperoleh gelar sarjana sains**

**MUHAMAD AULIA RAHMAN**

**0906602124**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
PROGRAM STUDI FISIKA EKSTENSI  
DEPOK  
DESEMBER 2011**

## HALAMAN PERNYATAAN ORISINALITAS

Skripsi ini adalah hasil karya saya sendiri,  
dan semua sumber baik yang dikutip maupun dirujuk  
telah saya nyatakan dengan benar.

Nama : Muhamad Aulia Rahman

NPM : 0906602124

Tanda Tangan : 

Tanggal : 22 Desember 2011

## HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :

Nama : Muhamad Aulia Rahman  
NPM : 0906602124  
Program studi : Instrumentasi  
Judul : Rancang Bangun Hotplate Stirrer Magnetik Terkendali  
Temperatur dan Kecepatan Pengaduk

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Muda pada Program Studi Instrumentasi Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Indonesia.

### DEWAN PENGUJI

Pembimbing : Dr. Bambang Soegijono M.Si (  )  
Pembimbing : Drs. Arief Sudarmaji, M.T (  )  
Penguji : Dr. Sastra Kusumawijaya (  )  
Penguji : Drs. Lingga. H, M.Si (  )

Ditetapkan di : Ruang Seminar Fakultas MIPA, Kampus UI Depok

Tanggal : 9 Desember 2011

## KATA PENGANTAR

Puji dan syukur kami panjatkan kepada Allah SWT beserta Nabi Muhammad SAW, yang telah melimpahkan segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan Laporan Tugas Akhir ini. Walaupun dalam penyusunan Laporan Tugas Akhir ini penulis menemukan berbagai macam kesulitan, tetapi Allah SWT senantiasa memberikan tetesan rahmat-Nya sehingga semua rintangan dan tantangan dapat dilalui dengan ridha-Nya.

Penyusunan Laporan Tugas Akhir yang berjudul “ **Rancang Bangun Hotplate Stirrer Magnetic Terkendali Temperatur dan Kecepatan Pengaduk** ” yang bertujuan untuk memenuhi syarat dalam menyelesaikan pendidikan program studi Ekstensi Instrumentasi, Departemen Fisika, FMIPA, Universitas Indonesia.

Dalam melaksanakan Tugas Akhir sampai penyelesaian Laporan Tugas Akhir ini, penulis banyak mendapat bantuan dan bimbingan dari berbagai pihak. Untuk itu pada kesempatan ini penulis ingin menyampaikan terima kasih dan rasa hormat kepada:

1. Allah SWT atas segala rahmat dan karunia-Nya yang berlimpah tanpa henti yang telah memberikan secercah pengetahuan serta ilmu yang bermanfaat.
2. Kedua orangtua dan Saudaraku untuk kasih sayang, doa dan dukungannya sehingga laporan ini selesai dibuat.
3. Prof. Dr. Drs. Terry Mart, selaku pembimbing akademis program Ekstensi Instrumentasi Elektronika Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Indonesia
4. Sastra Kusuma Wijaya Ph.D selaku Ketua peminatan S1 program Ekstensi Instrumentasi Elektronika Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Indonesia.
5. Dr. Bambang Soegijono M.Si, dan Drs. Arief Sudarmaji, M.T, selaku dosen pembimbing yang telah memberikan petunjuk, kemudahan dalam berpikir dan bimbingan dalam penyelesaian tugas akhir ini. Terima kasih

atas bimbingan dan kerjasamanya selama proses penelitian sehingga penelitian berjalan baik dan menyenangkan.

6. Seluruh dosen dan staff pengajar Program Ekstensi Instrumentasi Elektronika Departemen Fisika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Indonesia.
7. Pak Parno yang selalu membantu dan memudahkan dalam peminjaman alat selama penelitian di Laboratorium Elektronika dan Bengkel Mekanik.
8. Semua pihak yang secara tidak langsung ikut terlibat dalam pembuatan tugas akhir ini yang tidak saya sebutkan satu persatu, semoga amal baik yang telah dilakukan dibalas oleh ALLAH SWT.

Menyadari keterbatasan pengalaman dan kemampuan yang dimiliki penulis, sudah tentu terdapat kekurangan serta kemungkinan jauh dari sempurna, untuk itu penulis tidak menutup diri dan mengharapkan adanya saran serta kritik dari berbagai pihak yang sifatnya membangun guna menyempurnakan penyusunan tugas akhir ini.

Akhir kata semoga penyusunan tugas akhir ini dapat memberikan manfaat bagi semua pihak yang bersangkutan, khususnya bagi penulis dan umumnya bagi para pembaca.

Depok, Desember 2011

Penulis

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI  
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

---

---

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : Muhamad Aulia Rahman  
NPM : 0906602124  
Program Studi : Instrumentasi  
Departemen : Fisika  
Fakultas : Matematika dan Ilmu Pengetahuan Alam  
Jenis karya : Skripsi

demikian pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty-Free Right*)** atas karya ilmiah saya yang berjudul :

**Rancang Bangun Hotplate Stirrer Magnetik Terkendali Temperatur dan Kecepatan Pengaduk.**

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada tanggal : 22 Desember 2011

Yang menyatakan



( Muhamad Aulia Rahman )

## ABSTRAK

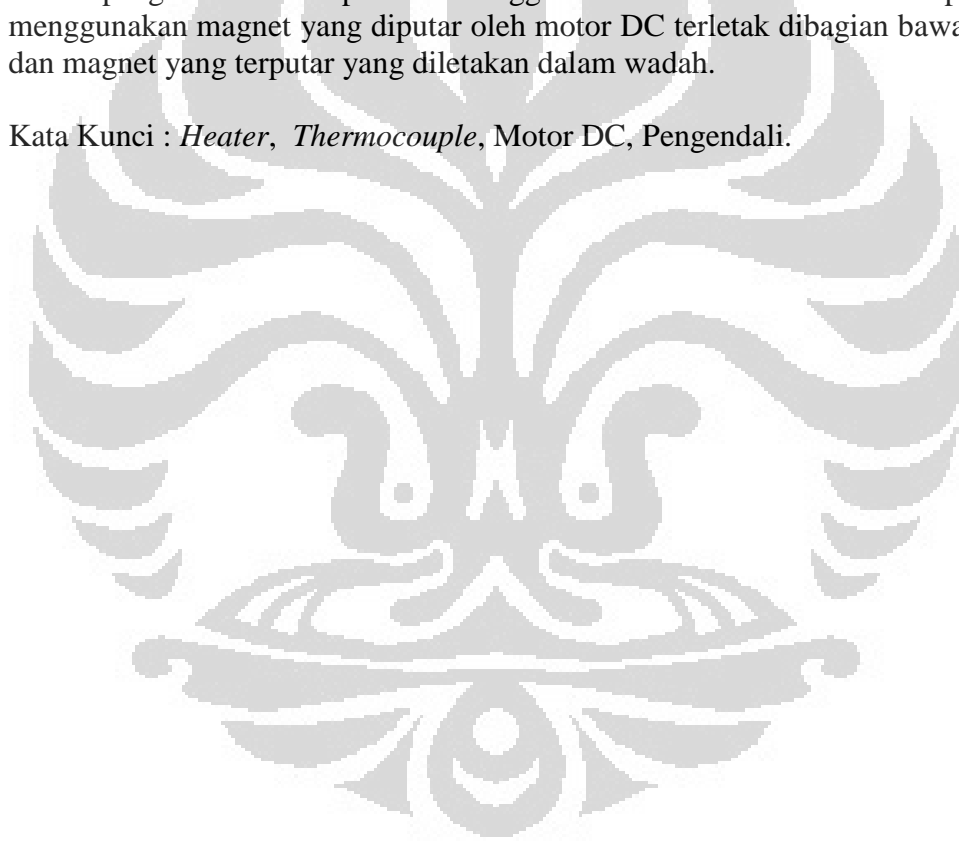
Nama : Muhamad Aulia Rahman

NPM : 0906602124

Judul : Rancang Bangun Hotplate Stirrer magnetik Terkendali Temperatur dan Kecepatan Pengaduk

Telah dibuat *Hotplate stirrer magnetik* dengan Temperatur dan Kecepatan Pengaduk Terkendali dengan menggunakan *Microcontroller*. Pada sistem ini menggunakan heater 500 watt, dan sensor Termokopel tipe K. Pengendalian motor DC menggunakan *sistem open loop* dan menggunakan *sistem close loop* untuk pengendalian temperatur menggunakan PID. Dalam sistem pangaduk menggunakan magnet yang diputar oleh motor DC terletak dibagian bawah heater dan magnet yang terputar yang diletakan dalam wadah.

Kata Kunci : *Heater, Thermocouple, Motor DC, Pengendali.*





## ABSTRACT

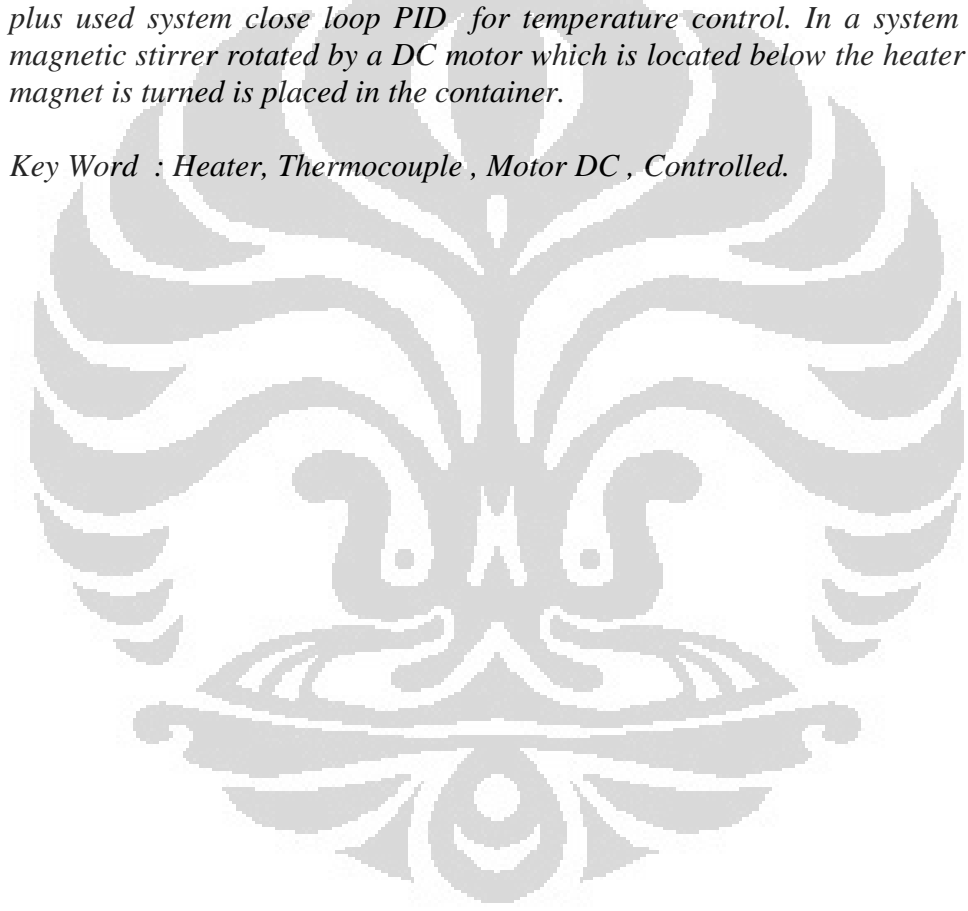
Nama : Muhamad Aulia Rahman

NPM : 0906602124

Judul : *Design Of a Controlled HotPlate Stirrer Magnetic Temperature and Speed Mixer*

*A Hotplate stirrer magnetic have been made with a high temperature and high speed restraint with microcontroller. This system have been used heater 500 watt, and thermocouple sensor type K. For DC motor control used system open loop plus used system close loop PID for temperature control. In a system using a magnetic stirrer rotated by a DC motor which is located below the heater and the magnet is turned is placed in the container.*

*Key Word : Heater, Thermocouple , Motor DC , Controlled.*



## DAFTAR ISI

|   |           |
|---|-----------|
| HALAMAN JUDUL.....  | i         |
| HALAMAN PERNYATAAN ORISINALITAS.....                                | ii        |
| HALAMAN PENGESAHAN.....   | iii       |
| KATA PENGANTAR .....  | iv        |
| ABSTRAK .....   | vii       |
| DAFTAR ISI.....   | ix        |
| DAFTAR GAMBAR .....   | xi        |
| DAFTAR TABEL .....  | xii       |
| DAFTAR LAMPIRAN.....  | xiii      |
| <b>BAB 1. PENDAHULUAN .....</b>                                     | <b>1</b>  |
| 1.1. Latar Belakang .....   | 1         |
| 1.2. Tujuan Penelitian .....  | 1         |
| 1.3. Deskripsi Singkat.....   | 2         |
| 1.4. Batasan Masalah .....  | 3         |
| 1.5. Metode Penelitian .....  | 3         |
| 1.6. Sistematika Penulisan .....                                    | 4         |
| <b>BAB 2. TEORI DASAR .....</b>                                     | <b>6</b>  |
| 2.1. <i>Thermocouple</i> .....                                      | 7         |
| 2.1.1 Prinsip Operasi <i>Thermocouple</i> .....                     | 7         |
| 2.1.2 Tipe-Tipe <i>Thermocouple</i> .....                           | 8         |
| 2.1.3 Penggunaan <i>Thermocouple</i> .....                          | 10        |
| 2.2. <i>Heater</i> (Pemanas) .....                                  | 10        |
| 2.3. Motor DC .....   | 11        |
| 2.3.1 Prinsip Dasar Cara Kerja .....                                | 12        |
| 2.3.2 Prinsip Arah Putaran Motor .....                              | 15        |
| 2.4. PWM (Pulse Width Modulation) .....                             | 16        |
| 2.5. Magnet .....   | 17        |
| 2.6. Teori Kontrol Proporsional Integral Differensial ( PID ) ..... | 18        |
| 2.6.1 Pengendali Proporsional (P).....                              | 22        |
| 2.6.2 Pengendali Integral (I).....                                  | 23        |
| 2.6.3 Pengendali Differensial (D) .....                             | 24        |
| <b>BAB 3. PERANCANGAN DAN CARA KERJA SISTEM .....</b>               | <b>26</b> |
| 3.1. Sistem kerja Alat.....   | 26        |
| 3.2. Perancangan Perangkat Keras ( Hardware ).....                  | 28        |
| 3.2.1 Rangkaian Minimum Sistem Atmega16 .....                       | 28        |
| 3.2.2 Konsep I/O pada Mikrocontroller AVR Atmega16.....             | 29        |
| 3.2.3 Konfigurasi Pin .....   | 30        |
| 3.2.4 Rangkaian Driver Motor .....                                  | 31        |
| 3.2.5 Rangkaian Cold Junction .....                                 | 32        |
| 3.2.6 Rangkaian Keypad .....  | 33        |

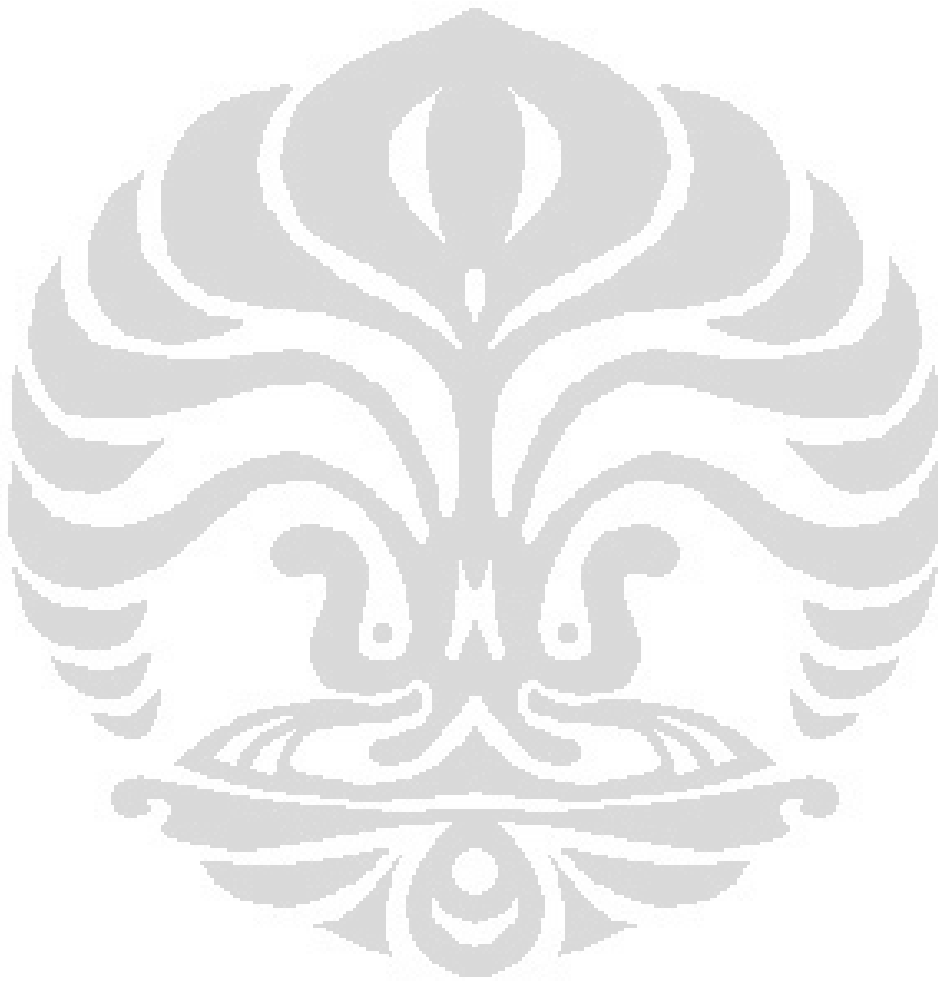
|  |           |
|--|-----------|
| 3.3. Perancangan Perangkat Lunak ( <i>Soft Ware</i> ).....                               | 34        |
| 3.3.1 Pengambilan Data .....   | 34        |
| 3.3.2 Proses pada sistem minimum Atmega16 .....  | 34        |
| 3.4. Perancangan Mekanik .....   | 37        |
| 3.4.1 Perancangan Heater.....  | 37        |
| 3.4.2 Perancangan Kotak Komponen dan motor DC.....                                       | 38        |
| 3.4.3 Perancangan Peredam Panas .....  | 39        |
| <b>BAB 4. HASIL PERCOBAAN DAN ANALISA .....</b>  | <b>42</b> |
| 4.1. Pengujian Rangkaian Minimum system dan keypad .....                                 | 42        |
| 4.2. Pengujian ADC (Analog to Digital Conversion).....                                   | 43        |
| 4.3. Pengkalibrasian Rangkaian Cold Junction.....  | 44        |
| 4.4. Pengujian <i>Thermocouple</i> .....   | 46        |
| 4.5. Pengujian Rangkaian Driver Motor .....  | 48        |
| 4.5.1 Pengujian Driver Motor .....   | 48        |
| 4.5.2 Pengujian Kecepatan Motor.....   | 48        |
| 4.5. Pengujian Sistem Pengendalian Dengan Metode <i>Direct</i><br><i>Synthesis</i> ..... | 50        |
| <b>BAB 5. KESIMPULAN DAN SARAN .....</b>   | <b>54</b> |
| 5.1. Kesimpulan .....  | 54        |
| 5.2. Saran.....  | 54        |
| <b>DAFTAR PUSTAKA .....</b>  | <b>55</b> |
| <b>LAMPIRAN</b>  |           |

## DAFTAR GAMBAR

|              |  |    |
|--------------|--|----|
| Gambar 1.1.  | Bagan keseluruhan sistem. ....   | 1  |
| Gambar 2.1.  | Rangkaian Thermocouple ( jenis besi-Constantan) .....                              | 7  |
| Gambar 2.2   | Output Thermocouple untuk jenis kawat yang berbeda.....                            | 8  |
| Gambar 2.3.  | Bentuk Fisik Thermocouple yang Digunakan.....                                      | 10 |
| Gambar 2.4.  | Heater jenis tubular .....   | 11 |
| Gambar 2.5.  | Motor D.C Sederhana.....   | 12 |
| Gambar 2.6.  | Medan magnet yang membawa arus mengelilingi konduktor .....                        | 12 |
| Gambar 2.7.  | Perubahan Medan magnet yang membawa arus .....                                     | 13 |
| Gambar 2.8.  | Medan magnet mengelilingi konduktor dan diantara kutub.....                        | 13 |
| Gambar 2.9.  | Reaksi garis fluks .....   | 13 |
| Gambar 2.10. | Prinsip kerja motor dc .....   | 14 |
| Gambar 2.11. | Sinyal PWM dengan duty cycle 50 %.....   | 16 |
| Gambar 2.12. | Sistem pengendali loop terbuka .....   | 19 |
| Gambar 2.13. | Sistem pengendali loop tertutup.....   | 20 |
| Gambar 2.14. | Blok diagram dari system control close loop.....                                   | 20 |
| Gambar 2.15. | Blok diagram pengendali proportional.....  | 23 |
| Gambar 2.16. | Grafik respon pengendali proporsional .....  | 24 |
| Gambar 2.17. | Blok diagram pengendali integral .....   | 24 |
| Gambar 2.18. | Grafik respon pengendali integral .....  | 25 |
| Gambar 2.19. | Blok diagram pengendali differensial .....   | 25 |
| Gambar 2.20. | Grafik respon pengendali differensial.....   | 26 |
| Gambar 3.1.  | Blok diagram cara kerja alat.....  | 27 |
| Gambar 3.2.  | Blok diagram pengendali temperatur .....   | 28 |
| Gambar 3.3.  | Rangkaian minimum sistem ATmega16.....   | 30 |
| Gambar 3.4.  | Pin-pin ATmega 16 kemasan 40-pin.....  | 31 |
| Gambar 3.5.  | Rangkaian driver motor.....  | 33 |
| Gambar 3.6.  | Rangkaian cold junction.....   | 34 |
| Gambar 3.7.  | Rangkaian dasar keypad.....  | 34 |
| Gambar 3.8.  | Flowchart Program Pengendali .....   | 37 |
| Gambar 3.9.  | Konsep pembuatan mekanik .....   | 38 |
| Gambar 3.10. | Perancangan heater.....  | 39 |
| Gambar 3.11. | Perancangan Kotak.....   | 40 |
| Gambar 3.12. | Peredam panas.....   | 41 |
| Gambar 3.13. | Keseluruhan Alat .....   | 41 |
| Gambar 4.1.  | Perubahan nilai ADC terhadap tegangan input.....                                   | 44 |
| Gambar 4.2.  | Grafik rata-rata temperature turun.....  | 47 |
| Gambar 4.3.  | Perubahan temperatur pada daya 70% .....   | 49 |
| Gambar 4.4.  | Respon kecepatan motor terhadap PWM.....   | 50 |
| Gambar 4.5.  | Mencari Nilai $\theta$ dan Nilai $\tau$ .....                                      | 52 |
| Gambar 4.6.  | Respon Temperatur terhadap waktu pada set point 240°C.....                         | 53 |
| Gambar 4.7.  | Respon Temperatur terhadap waktu pada <i>set point</i> 110°C .....                 | 53 |
| Gambar 4.8.  | Perbandingan respon system dengan pengendalian PID dan tanpa pengendalian PID..... | 54 |

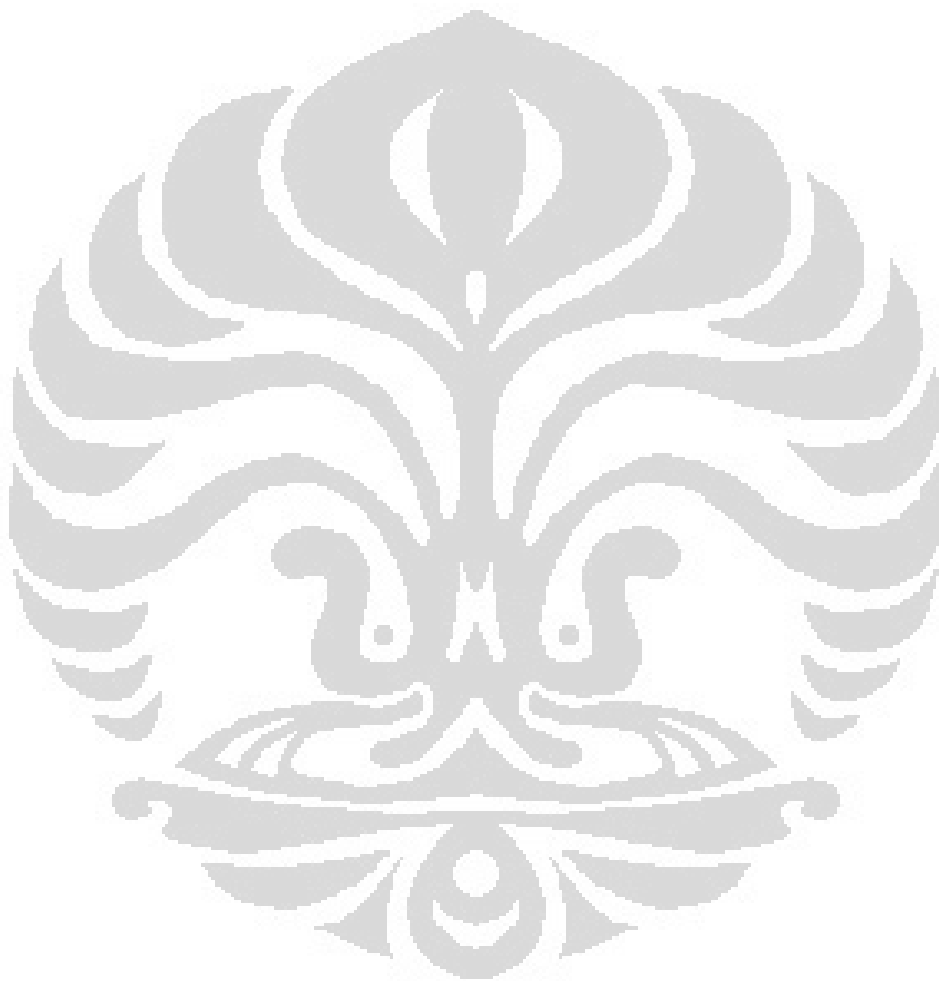
## DAFTAR TABEL

|   |    |
|---|----|
| Tabel 2.1. Jenis-jenis heater .....                               | 11 |
| Tabel 4.1 Nilai keypad yang ditampilkan pada LCD.....             | 43 |
| Tabel 4.2. Data pengujian ADC .....                               | 44 |
| Tabel 4.3. Data hasil pengujian kecepatan motor terhadap PWM..... | 49 |



## DAFTAR LAMPIRAN

- Lampiran 1 : Rangkaian system Alat
- Lampiran 2 : Data kalibrasi *thermocouple*
- Lampiran 3 : Program Bascom



# BAB 1

## PENDAHULUAN

Bab ini menjelaskan mengenai latar belakang masalah mengapa alat ini dibuat, tujuan penelitian, deskripsi singkat mengenai alat yang akan dibuat, batasan masalah dari alat yang akan dibuat oleh penulis, metodologi penelitian, dan sistematika penulisan laporan.

### 1.1 Latar Belakang

Pada era di globalisasi ini semakin berkembang teknologi di segala bidang baik komunikasi, industri, pendidikan, kesehatan, dll maka semakin banyak alat yang dibuat/ dirancang dan diciptakan dengan maksud agar dapat mempermudah aktivitas manusia dalam melaksanakan segala macam kegiatannya di berbagai bidang.

Kemajuan teknologi berkembang dengan pesat seperti dapat dilihat dalam katalog produk alat laboratorium dan penunjang pendidikan yang digunakan semakin canggih tetapi yang menjadi kendalanya adalah harga alat-alat tersebut yang relatif mahal. Oleh karena hal itu saya mencoba untuk membuat suatu alat yang dapat digunakan pada laboratorium kimia dengan harga yang relatif lebih terjangkau. Hal ini dilakukan untuk memperlihatkan bahwa untuk mendapatkan suatu alat yang canggih tidak dibutuhkan biaya yang terlalu tinggi.

Alat yang saya coba buat ini dinamakan ” **Rancang Bangun Hotplate Stirrer Magnetik terkendali temperatur dan kecepatan Pengaduk**”. Pada alat ini sistem yang akan dikendalikan adalah sistem temperatur dan kecepatan dari pengaduknya. Pengendalian sistem ini dengan menggunakan *keypad* yang kemudian data dihubungkan dengan rangkaian *microcontroller* untuk diproses dan dikirim ke *actuator* sehingga didapatkan hasil yang diinginkan.

### 1.2 Tujuan Penelitian

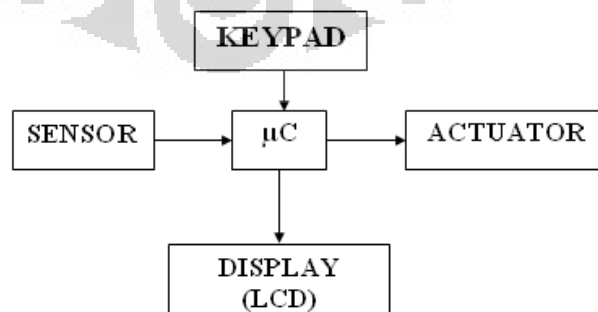
Tujuan membuat rancang bangun hotplate stirrer ini adalah untuk membuat suatu alat pencampur atau pemanas dengan temperatur dan kecepatan pengaduknya yang terkendali.

### 1.3 Deskripsi Singkat

Sebagai salah satu aplikasi teknologi pada dunia kesehatan, pada tugas akhir ini dibuat “**Rancang Bangun Hotplate Stirrer Magnetik Terkendali Temperatur dan Kecepatan Pengaduk**” dengan maksud menekan pembiayaan dan memudahkan pekerja dalam bidang kesehatan sehingga proses kerja yang dilakukan dapat berjalan dengan mudah, aman dan efisien. Hal tersebut dikarenakan pengontrolan kerja untuk kenaikan temperatur dan kecepatan pengaduk alat yang dilakukan dengan memasukan data input menggunakan *keypad*.

Cara kerja alat ini adalah pada saat ingin menghasilkan temperatur yang cukup tinggi, maka tombol temperatur ditekan untuk diatur sesuai dengan keinginan sehingga terjadi proses pemanasan, yang sebelumnya telah diatur temperaturnya. Begitu pula untuk gerakan kecepatan pengaduknya, kecepatan yang dihasilkan akan sesuai dengan kecepatan yang telah diatur melalui *keypad*. Hal tersebut disebabkan oleh data dari *keypad* tersebut menginisialisasikan *microcontroller* yang bertindak sebagai pengendali. Perubahan temperatur, kecepatan pengaduk ataupun waktu kerja alat dapat dilihat pada *display*. Dan sensor menunjukkan temperatur yang akan diatur yang kemudian ditampilkan pada *display*.

Sistem ini terdiri dari sebuah sensor temperatur, sebuah *microcontroller* sebagai pengolah data, *actuator* berupa motor sebagai pemutar pengaduk, magnet sebagai pengaduk, dan *heater* yang dirancang berbentuk *hot plate* sebagai pemanas dan *display* hasil dari pendeteksian tersebut. Berikut ini adalah bagian dari keseluruhan sistem :



Gambar 1.1. Bagan Keseluruhan Sistem



#### 1.4 Batasan Masalah

Pembatasan masalah ini menitik beratkan pada sistem pengendali temperatur dan kecepatan motor pada stirrer. Dimana yang akan dibahas adalah segala proses yang berhubungan dengan sistem pengendalian dengan menggunakan program.

#### 1.5 Metode Penelitian

Metode yang dilakukan untuk membantu dalam pelaksanaan dan penganalisaan alat ini:

1. Studi Literatur

Penulis menggunakan metode ini untuk memperoleh informasi yang berkaitan dengan penelitian yang penulis buat. Study literatur ini mengacu pada buku-buku pegangan, data sheet dari berbagai macam komponen yang di pergunakan, data yang didapat dari internet, dan makalah-makalah yang membahas tentang proyek yang penulis buat.

2. Perancangan dan Pembuatan Alat

Berisi tentang proses perancangan alat berupa sistem pengendali berbasis microcontroller dan mekanik. Pada bagian sistem pengendali akan membahas masalah pembuatan minimum sistem *controller*.

3. Pembuatan Program

Tahap ini merupakan proses pembuatan program dilakukan dengan menggunakan *Software Basic Compiler (BASCOM)*, dengan menggunakan *Software* ini memungkinkan kita untuk memanipulasi kinerja alat sesuai dengan yang diinginkan.

4. Uji Sistem

Dari alat yang dibuat maka dilakukan pengujian terhadap masing-masing bagian dengan tujuan untuk mengetahui kinerjanya agar sesuai dengan apa yang diharapkan dan dapat melakukan pengambilan data.

5. Pengambilan Data

Pada bab ini akan diuraikan tentang kinerja dari masing-masing blok

data yang diambil dengan harapan dalam pengujian tidak terdapat kesalahan yang fatal.

#### 6. Penulisan Penelitian

Dari hasil pengujian dan pengambilan data kemudian dilakukan suatu analisa sehingga dapat diambil suatu kesimpulan. Dengan adanya beberapa saran juga dapat kita ajukan sebagai bahan perbaikan untuk penelitian lebih lanjut.

### 1.6 Sistematika Penulisan

Sistematika penulisan Tugas Akhir ini terdiri dari bab-bab yang memuat beberapa sub-bab. Untuk memudahkan pembacaan dan pemahaman maka tugas akhir ini dibagi menjadi beberapa bab yaitu:

#### **BAB 1 PENDAHULUAN**

Pendahuluan berisi latar belakang, tujuan penelitian, deskripsi singkat, pembatasan masalah, metode penulisan dan sistematika penulisan dari tugas akhir ini.

#### **BAB 2 TEORI DASAR**

Teori dasar berisi landasan-landasan teori sebagai hasil dari studi literatur yang berhubungan dalam perancangan dan pembuatan alat.

#### **BAB 3 PERANCANGAN DAN CARA KERJA SISTEM**

Pada bab ini akan dijelaskan secara keseluruhan tentang sistem pengendalian temperatur dan kecepatan pengaduk pada *stirrer*.

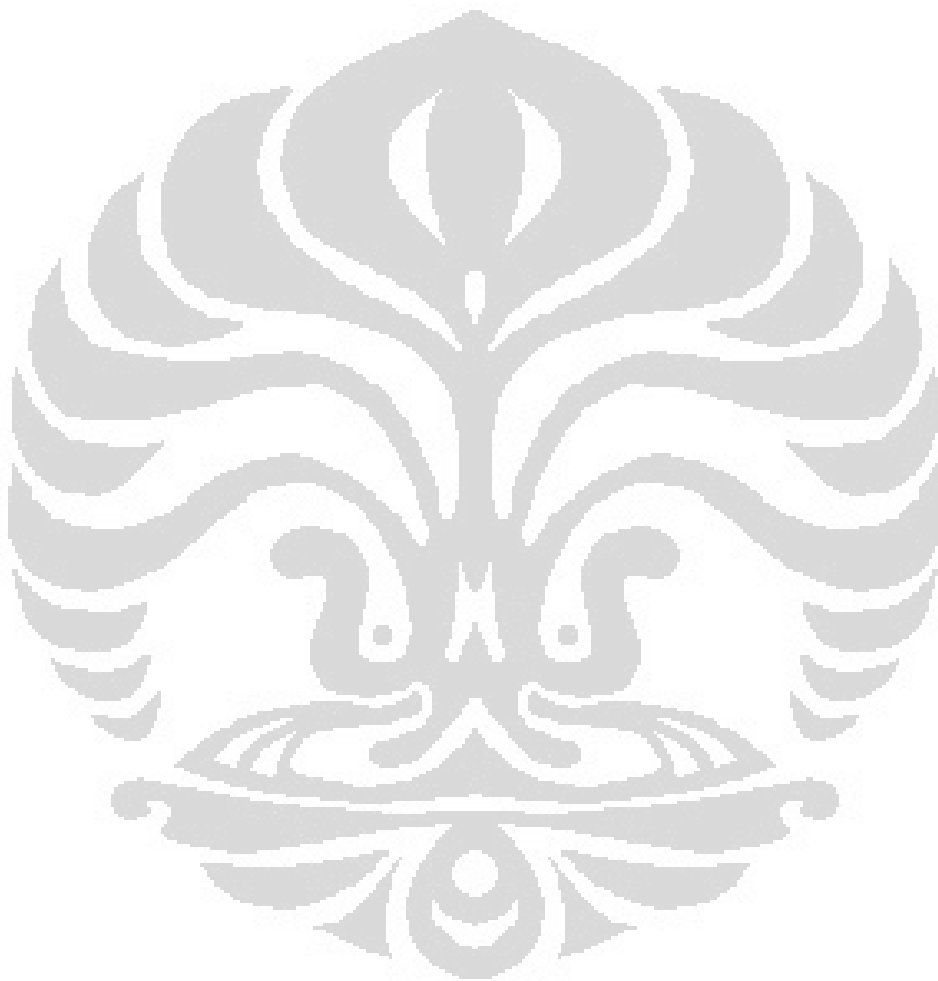
#### **BAB 4 DATA PERCOBAAN DAN ANALISA**

Bab ini berisi tentang unjuk kerja alat sebagai hasil dari perancangan sistem. Pengujian akhir dilakukan dengan menyatukan seluruh bagian-bagian kecil dari sistem untuk memastikan bahwa sistem dapat berfungsi sesuai dengan tujuan awal. Setelah sistem berfungsi dengan baik maka

dilanjutkan dengan pengambilan data untuk memastikan kapabilitas dari sistem yang dibangun.

## **BAB 5 KESIMPULAN DAN SARAN**

Penutup berisi kesimpulan yang diperoleh dari pengujian sistem dan pengambilan data selama penelitian berlangsung, selain itu juga penutup memuat saran untuk pengembangan lebih lanjut dari penelitian ini.



## BAB 2

### TEORI DASAR

Pada sistem pengendali *hotplate stirrer*, diperlukan beberapa pemahaman dasar. Beberapa pemahaman dasar tersebut antara lain: prinsip dasar *heater* (pemanas), sensor temperatur (*Thermocouple*), motor DC, PWM (*Pulse Width Modulation*), komunikasi data serial, dan PID.

#### 2.1 *Thermocouple*

*Thermocouple* dikembangkan lebih dari 100 tahun yang lalu dan masih suka digunakan secara luas, terutama disituasi suhu tinggi. *Thermocouple* adalah sensor suhu yang banyak digunakan untuk mengubah perbedaan panas dalam benda yang diukur temperaturnya menjadi perubahan potensial atau tegangan listrik. *Thermocouple* yang sederhana dapat dipasang, dan memiliki jenis konektor standar yang sama, serta dapat mengukur temperatur dalam jangkauan suhu yang cukup besar dengan batas kesalahan pengukuran kurang dari 1 °C.

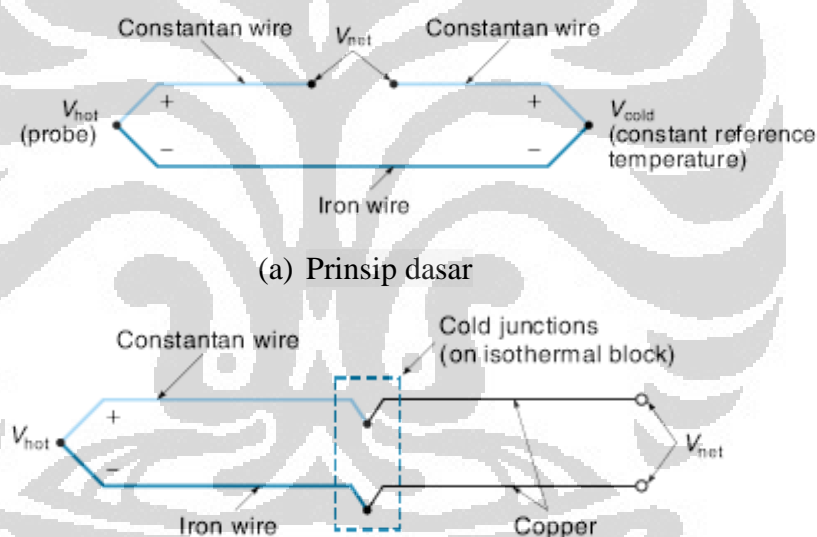
##### 2.1.1 Prinsip Operasi *Thermocouple*

*Thermocouple* ini didasarkan pada efek seebeck, fenomena dimana tegangan yang proporsional dengan suhu dapat dihasilkan dari rangkaian yang terdiri dari dua kawat logam yang berbeda. Sebagai contoh, sebuah *Thermocouple* terbuat dari besi dan Constanta (paduan) menghasilkan tegangan sekitar 35  $\mu\text{V}/^\circ\text{F}$ . Gambar 2.1. (a) menggambarkan situasi ini. Kita bisa berpikir dari sambungan di setiap akhir kawat logam berbeda sebagai penghasil tegangan, sehingga tegangan bersih ( $V_{\text{net}}$ ) sebenarnya perbedaan Antara tegangan persimpangan. Salah satu persimpangan adalah pada probe dan disebut persimpangan panas [1]. Sambungan lainnya disimpan di beberapa suhu acuan dikenal dan disebut persimpangan dingin, atau sambungan referensi. Tegangan output dari sistem ini dapat dinyatakan sebagai berikut:

$$V_{\text{net}} = V_{\text{hot}} - V_{\text{cold}} \quad (2.1)$$

Dalam prakteknya, kabel Thermocouple harus terhubung ke kabel tembaga di beberapa titik seperti yang ditunjukkan pada Gambar 2.1. (b), sehingga sesungguhnya ada tiga persimpangan. Namun, ternyata tegangan total dari dua persimpangan tembaga akan sama dengan tegangan dingin persimpangan tunggal ( $V_{\text{cold}}$ ) Gambar 2.1. (a) (dengan asumsi persimpangan tembaga pada suhu yang sama), sehingga analisis yang tidak berubah.

Secara tradisional, persimpangan dingin disimpan pada  $32^\circ\text{F}$  dalam tempat air es, yang di dalamnya terdapat air es. Air es ini digunakan karena merupakan salah satu cara untuk menghasilkan suhu yang dikenal, dan sebagainya  $V_{\text{cold}}$  menjadi konstan dalam Persamaan 2.2, meninggalkan hubungan langsung antara  $V_{\text{net}}$  dan  $V_{\text{hot}}$ :



(a) Prinsip dasar  
(b) Thermocouple dihubungkan dengan kabel tembaga  
Gambar 2.1. Rangkaian Thermocouple ( jenis besi-Constantan)[1]

$$V_{\text{hot}} = V_{\text{net}} + \text{constan} \quad (2.2)$$

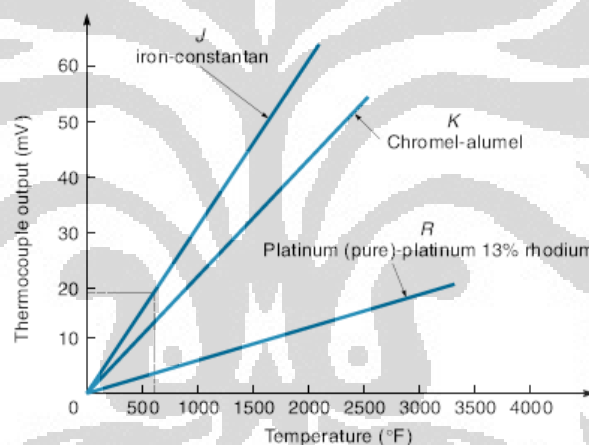
dimana :

constan =  $V_{\text{cold}}$  saat temperature referensi

Sistem modern menghilangkan kebutuhan untuk air es. Salah satu metode adalah untuk menjaga *cold junction* pada temperatur konstan dengan sistem kontrol. Hal ini dapat berguna jika ada banyak Thermocouple dalam sistem

semua bisa direferensikan ke suhu yang sama. Metode lain (digunakan oleh controller komputer) adalah dengan hanya melihat di tabel nilai  $V_{\text{cold}}$  untuk suhu ruang dan menambahkan nilai untuk  $V_{\text{net}}$  untuk menghasilkan  $V_{\text{hot}}$ .

*Thermocouple* komersial tersedia dengan rentang suhu dan kepekaan yang berbeda (sensitivitas menjadi ukuran volt / derajat). Gambar 2.2. menunjukkan kurva volt terhadap suhu pada kelas *Thermocouple*. Seperti yang Anda lihat, tipe J (besi-Constantan) memiliki sensitivitas tertinggi tetapi kisaran suhu terendah, tipe K (chromel-alumel) memiliki kisaran suhu yang lebih tinggi tetapi sensitivitas yang lebih rendah, dan tipe R (platinum-chodium) memiliki sensitivitas lebih rendah namun dapat bekerja pada suhu tinggi.



Gambar 2.2 Output Thermocouple untuk jenis kawat yang berbeda (direferensikan pada 32 ° F)[1]

### 2.1.2 Tipe-Tipe *Thermocouple*

Tersedia beberapa jenis *thermocouple*, tergantung aplikasi penggunaannya

1. Tipe K (Chromel (Ni-Cr alloy) / Alumel (Ni-Al alloy))

*Thermocouple* untuk tujuan umum. Lebih murah. Tersedia untuk rentang suhu  $-200\text{ }^{\circ}\text{C}$  hingga  $+1200\text{ }^{\circ}\text{C}$ .

2. Tipe E (Chromel / Constantan (Cu-Ni alloy))

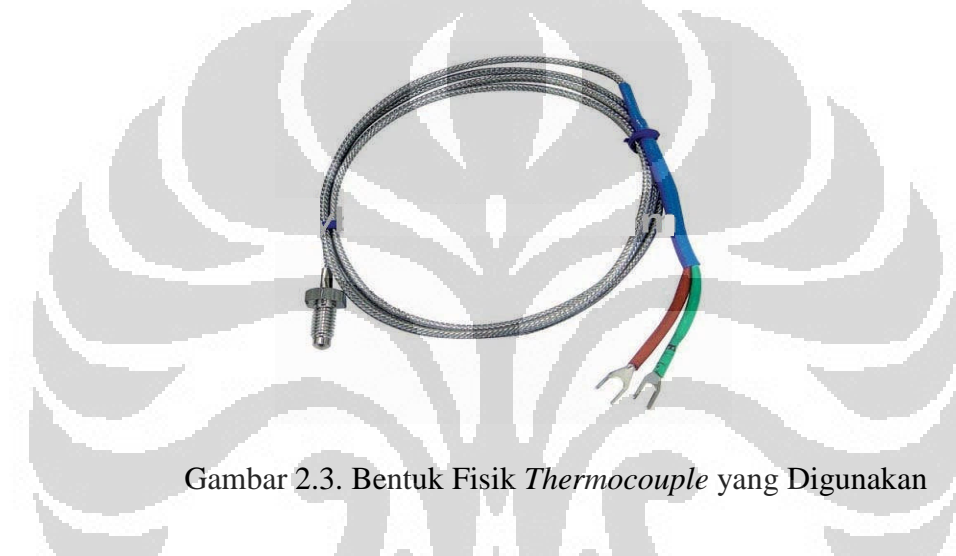
3. Tipe E memiliki output yang besar ( $68 \mu\text{V}/^\circ\text{C}$ ) membuatnya cocok digunakan pada temperatur rendah. Properti lainnya tipe E adalah tipe non magnetik.
4. Tipe J (Iron / Constantan) Rentangnya terbatas ( $-40$  hingga  $+750^\circ\text{C}$ ) membuatnya kurang populer dibanding tipe K
5. Tipe J memiliki sensitivitas sekitar  $52 \mu\text{V}/^\circ\text{C}$
6. Tipe N (Nicrosil (Ni-Cr-Si alloy) / Nisil (Ni-Si alloy)) Stabil dan tahanan yang tinggi terhadap oksidasi membuat tipe N cocok untuk pengukuran suhu yang tinggi tanpa platinum. Dapat mengukur suhu di atas  $1200^\circ\text{C}$ . Sensitifitasnya sekitar  $39 \mu\text{V}/^\circ\text{C}$  pada  $900^\circ\text{C}$ , sedikit di bawah tipe K. Tipe N merupakan perbaikan tipe K.
7. *Thermocouple* tipe B, R, dan S adalah *thermocouple* logam mulia yang memiliki karakteristik yang hampir sama. Mereka adalah *thermocouple* yang paling stabil, tetapi karena sensitifitasnya rendah (sekitar  $10 \mu\text{V}/^\circ\text{C}$ ) mereka biasanya hanya digunakan untuk mengukur temperatur tinggi ( $>300^\circ\text{C}$ ).
8. Tipe B (Platinum-Rhodium/Pt-Rh) Cocok mengukur suhu di atas  $1800^\circ\text{C}$ . Tipe B memberi output yang sama pada suhu  $0^\circ\text{C}$  hingga  $42^\circ\text{C}$  sehingga tidak dapat dipakai di bawah suhu  $50^\circ\text{C}$ .
9. Tipe R (Platinum /Platinum with 7% Rhodium) Cocok mengukur suhu di atas  $1600^\circ\text{C}$ . sensitivitas rendah ( $10 \mu\text{V}/^\circ\text{C}$ ) dan biaya tinggi membuat mereka tidak cocok dipakai untuk tujuan umum.
10. Tipe S (Platinum /Platinum with 10% Rhodium) Cocok mengukur suhu di atas  $1600^\circ\text{C}$ . sensitivitas rendah ( $10 \mu\text{V}/^\circ\text{C}$ ) dan biaya tinggi membuat mereka tidak cocok dipakai untuk tujuan umum. Karena stabilitasnya yang tinggi Tipe S digunakan untuk standar pengukuran titik leleh emas ( $1064.43^\circ\text{C}$ ).
11. Tipe T (Copper / Constantan) Cocok untuk pengukuran antara  $-200$  to  $350^\circ\text{C}$ . Konduktor positif terbuat dari tembaga, dan yang negatif terbuat dari constantan. Sering dipakai sebagai alat pengukur alternatif sejak penelitian kawat tembaga. Tipe T memiliki sensitifitas  $43 \mu\text{V}/^\circ\text{C}$ .

### 2.1.3 Penggunaan *Thermocouple*

*Thermocouple* paling cocok digunakan untuk mengukur rentangan suhu yang luas, hingga 1800 K. Dalam pembuatan alat ini menggunakan *thermocouple* tipe K, dimana *thermocouple* memiliki karakteristik sebagai berikut, yaitu :

- a. Terbuat dari bahan Chromel (Ni-Cr) dan Alumel (Ni-Al)
- b. Aktif pada suhu  $-200^{\circ}\text{C} \sim 1200^{\circ}\text{C}$
- c. Sensitivitasnya @ $25^{\circ}\text{C}$  adalah  $40,6\mu\text{V}/^{\circ}\text{C}$

Berikut adalah respon tegangan keluaran *thermocouple* terhadap temperature berdasarkan tipe *thermocouple*.



Gambar 2.3. Bentuk Fisik *Thermocouple* yang Digunakan

## 2.2 *Heater* ( Pemanas )

*Heater* adalah suatu alat yang digunakan untuk memancarkan panas atau suatu alat yang digunakan untuk mencapai temperatur yang lebih tinggi. Didalam istilah elektronika, *heater* adalah kumpulan dari kawat serabut yang terdapat didalam ruang hampa udara yang berfungsi untuk memanaskan katode didalam suatu termisi emisi electron.

*Heater* dapat digunakan sebagai pemanas langsung yang dapat mengakibatkan reaksi-reaksi tertentu. Misalnya, di bidang industri kimia atau untuk menghasilkan panas dalam reaksi kimia untuk proses seperti memecah. *Water heater* adalah suatu proses thermodynamic dengan menggunakan suatu sumber energi untuk memanaskan air diatas temperatur awalnya. Bentuk secara fisik heater yang digunakan penulis dapat dilihat dari Gambar dibawah ini. <sup>[2]</sup>





Gambar 2.4. Heater jenis tubular

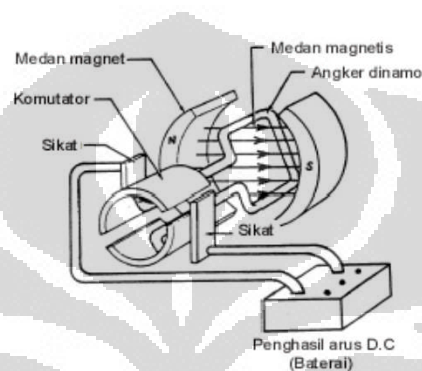
**Tabel 2.1.** Jenis-jenis heater [2]

| Jenis Heater                | Sifat Benda yang Dipanaskan | Memaskan / Membuat   |
|-----------------------------|-----------------------------|--|
| Tubular Straight, Multiform | Padat                       | Direkatkan pada dies, heat sealing tools, dll.                   |
| Tubular Straight, Multiform | Cair                        | Air, minyak, plating, aspal, garam, dll                          |
| Tubular                     | Permukaan benda Padat       | Drying, baking, kain, plastic, makanan, dll.                     |
| Immersion Heater            | Cair                        | Air, minyak, plating, aspal, garam, dll                          |
| Finned Heater               | Gas                         | Menghangatkan oven, ruangan, dll.                                |
| In – Line                   | Cair, Gas                   | Air, memanaskan minyak sebelum dikeluarkan ke mesin burner, dll. |

### 2.3 Motor DC

Motor listrik merupakan perangkat elektromagnetis yang mengubah energi listrik menjadi energi mekanik. Energi mekanik ini digunakan untuk, misalnya memutar impeller pompa, fan atau blower, menggerakkan kompresor, mengangkat bahan, dll. Motor listrik digunakan juga di rumah (mixer, bor listrik, fan angin) dan di industri. Motor listrik kadangkala disebut “kuda kerja” nya industri sebab diperkirakan bahwa motor-motor menggunakan sekitar 70% beban listrik total di industri. Motor DC memerlukan suplai tegangan yang searah pada kumparan medan untuk diubah menjadi energi mekanik. Kumparan medan pada motor dc disebut stator (bagian yang tidak berputar) dan kumparan jangkar disebut rotor (bagian yang berputar). Jika terjadi putaran pada kumparan jangkar dalam pada

medan magnet, maka akan timbul tegangan (GGL) yang berubah-ubah arah pada setiap setengah putaran, sehingga merupakan tegangan bolak-balik. Prinsip kerja dari arus searah adalah membalik fasa tegangan dari gelombang yang mempunyai nilai positif dengan menggunakan komutator, dengan demikian arus yang berbalik arah dengan kumparan jangkar yang berputar dalam medan magnet. Bentuk motor paling sederhana memiliki kumparan satu lilitan yang bisa berputar bebas di antara kutub-kutub magnet permanen. [3]

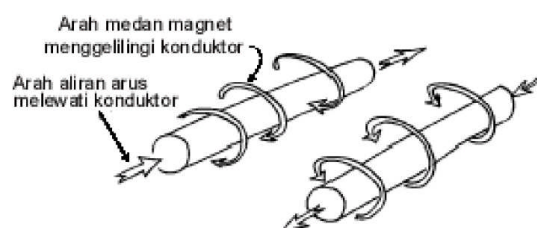


Gambar 2.5. Motor D.C Sederhana [3]

Catu tegangan dc dari baterai menuju ke lilitan melalui sikat yang menyentuh komutator, dua segmen yang terhubung dengan dua ujung lilitan. Kumparan satu lilitan pada gambar di atas disebut angker dinamo. Angker dinamo adalah sebutan untuk komponen yang berputar di antara medan magnet. [2]

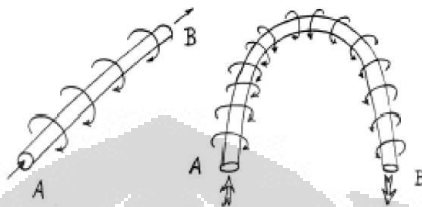
### 2.3.1 Prinsip Dasar Cara Kerja

Jika arus lewat pada suatu konduktor, timbul medan magnet di sekitar konduktor. Arah medan magnet ditentukan oleh arah aliran arus pada konduktor.



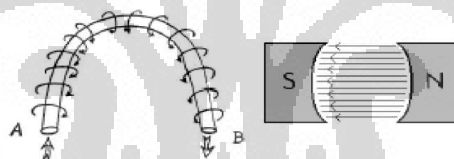
Gambar 2.6. Medan magnet yang membawa arus mengelilingi konduktor [3]

Aturan Genggaman Tangan Kanan bisa dipakai untuk menentukan arah garis fluks di sekitar konduktor. Genggam konduktor dengan tangan kanan dengan jempol mengarah pada arah aliran arus, maka jari-jari anda akan menunjukkan arah garis fluks. Gambar 2.6 menunjukkan medan magnet yang terbentuk di sekitar konduktor berubah arah karena bentuk U.



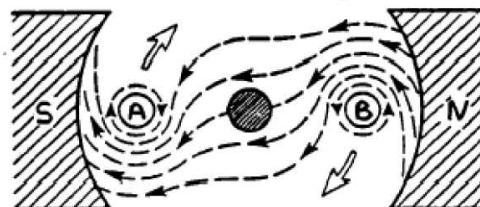
Gambar 2.7. Perubahan Medan magnet yang membawa arus mengelilingi konduktor [3]

Catatan : Medan magnet hanya terjadi di sekitar sebuah konduktor jika ada arus mengalir pada konduktor tersebut. Pada motor listrik konduktor berbentuk U disebut angker dinamo.



Gambar 2.8. Medan magnet mengelilingi konduktor dan diantara kutub [3]

Jika konduktor berbentuk U (angker dinamo) diletakkan di antara kutub utara dan selatan yang kuat medan magnet konduktor akan berinteraksi dengan medan magnet kutub. Lihat Gambar 2.6.

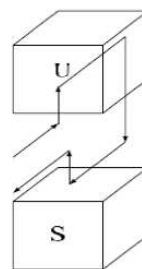


Gambar 2.9. Reaksi garis fluks [3]

Lingkaran bertanda A dan B merupakan ujung konduktor yang dilengkungkan (looped conductor). Arus mengalir masuk melalui ujung A dan keluar melalui ujung B. Medan konduktor A yang searah jarum jam akan menambah medan pada kutub dan menimbulkan medan yang kuat di bawah konduktor. Konduktor akan berusaha bergerak ke atas untuk keluar dari medan kuat ini. Medan konduktor B yang berlawanan arah jarum jam akan menambah medan pada kutub dan menimbulkan medan yang kuat di atas konduktor. Konduktor akan berusaha untuk bergerak turun agar keluar dari medan yang kuat tersebut. Gaya-gaya tersebut akan membuat angker dinamo berputar searah jarum jam. Mekanisme kerja untuk seluruh jenis motor secara umum :

- Arus listrik dalam medan magnet akan memberikan gaya.
- Jika kawat yang membawa arus dibengkokkan menjadi sebuah lingkaran / loop, maka kedua sisi loop, yaitu pada sudut kanan medan magnet, akan mendapatkan gaya pada arah yang berlawanan.
- Pasangan gaya menghasilkan tenaga putar / torque untuk memutar kumparan.
- Motor-motor memiliki beberapa loop pada dinamonya untuk memberikan tenaga putaran yang lebih seragam dan medan magnetnya dihasilkan oleh susunan elektromagnetik yang disebut kumparan medan.

Pada motor dc, daerah kumparan medan yang dialiri arus listrik akan menghasilkan medan magnet yang melingkupi kumparan jangkar dengan arah tertentu. Konversi dari energi listrik menjadi energi mekanik (motor) maupun sebaliknya berlangsung melalui medan magnet, dengan demikian medan magnet disini selain berfungsi sebagai tempat untuk menyimpan energi, sekaligus sebagai tempat berlangsungnya proses perubahan energi.



Gambar 2.10. Prinsip kerja motor dc [3]

Agar proses perubahan energi mekanik dapat berlangsung secara sempurna, maka tegangan sumber harus lebih besar daripada tegangan gerak yang disebabkan reaksi lawan. Dengan memberi arus pada kumparan jangkar yang dilindungi oleh medan maka menimbulkan perputaran pada motor.

Dalam memahami sebuah motor, penting untuk mengerti apa yang dimaksud dengan beban motor. Beban dalam hal ini mengacu kepada keluaran tenaga putar / torque sesuai dengan kecepatan yang diperlukan. Beban umumnya dapat dikategorikan ke dalam tiga kelompok :

- Beban torque konstan adalah beban dimana permintaan keluaran energinya bervariasi dengan kecepatan operasinya namun torquency tidak bervariasi. Contoh beban dengan torque konstan adalah corveyors, rotary kilns, dan pompa displacement konstan.
- Beban dengan variabel torque adalah beban dengan torque yang bervariasi dengan kecepatan operasi. Contoh beban dengan variabel torque adalah pompa sentrifugal dan fan (torque bervariasi sebagai kuadrat kecepatan).  
Peralatan Energi Listrik : Motor Listrik.
- Beban dengan energi konstan adalah beban dengan permintaan torque yang berubah dan berbanding terbalik dengan kecepatan. Contoh untuk beban dengan daya konstan adalah peralatan-peralatan mesin.

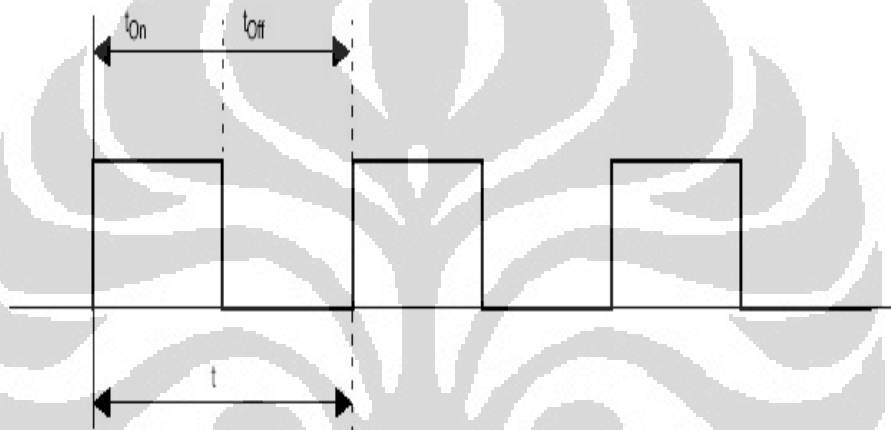
### 2.3.2 Prinsip Arah Putaran Motor

Untuk menentukan arah putaran motor digunakan kaedah Flamming tangan kiri. Kutub-kutub magnet akan menghasilkan medan magnet dengan arah dari kutub utara ke kutub selatan. Jika medan magnet memotong sebuah kawat penghantar yang dialiri arus searah dengan empat jari, maka akan timbul gerak searah ibu jari. Gaya ini disebut gaya Lorentz, yang besarnya sama dengan  $F$  [3].

Prinsip motor : aliran arus di dalam penghantar yang berada di dalam pengaruh medan magnet akan menghasilkan gerakan. Besarnya gaya pada penghantar akan bertambah besar jika arus yang melalui penghantar bertambah besar.

## 2.4 PWM (Pulse Width Modulation)

Suatu teknik yang digunakan untuk mengontrol kerja dari suatu alat atau menghasilkan suatu tegangan DC yang variabel adalah PWM (Pulse Width Modulation). Rangkaian PWM adalah rangkaian yang lebar pulsa tegangan keluarannya dapat diatur atau dimodulasi oleh sinyal tegangan modulasi. Disamping itu kita dapat menghasilkan suatu sinyal PWM dengan menentukan frekuensi dan waktu dari variabel ON dan OFF. Pemodulasian sinyal yang beragam dapat menghasilkan duty cycle yang diinginkan. Gambar 2.10 memperlihatkan sinyal kotak dengan duty cycle 50%. [4]



Gambar 2.11. Sinyal PWM dengan duty cycle 50 % [4]

Duty cycle (lihat persamaan 2.2) adalah ratio dari waktu ON ( $t_{ON}$ ) terhadap periode total dari sinyal ( $t=t_{ON} + t_{OFF}$ ).

Dengan persamaan :

$$D = \frac{t_1}{t_1 + t_2} \quad (2.1)$$

Dimana :  $D =$  Duty Cycle

$t_1 = t_{ON}$

$t_2 = t_{OFF}$

sehingga frekuensi yang dapat dihasilkan :

$$F = \frac{1}{T} \quad (2.2)$$

Dimana :  $T = \text{periode} = (t_1 + t_2)$

Modulasi lebar pulsa yang dihasilkan dari teknik PWM ini akan digunakan untuk mengatur kecepatan dari motor DC.

## 2.5 Magnet

Kemagnetan adalah suatu sifat zat yang teramati sebagai suatu gaya tarik atau gaya tolak antara kutub- kutub tidak senama maupun senama. Gaya magnet tersebut paling kuat di dekat ujung-ujung atau kutub-kutub magnet tersebut. Semua magnet memiliki dua kutub magnet yang berlawanan, utara (U) dan selatan (S). Apabila sebuah magnet batang digantung maka magnet tersebut berputar secara bebas, kutub utara akan menunjuk ke utara <sup>[5]</sup>. Magnet dapat menarik benda lain. Beberapa benda bahkan tertarik lebih kuat dari yang lain, yaitu bahan logam. Namun tidak semua logam mempunyai daya tarik yang sama terhadap magnet. Besi dan baja adalah dua contoh materi yang mempunyai daya tarik yang tinggi oleh magnet. Sedangkan oksigen cair adalah contoh materi yang mempunyai daya tarik yang rendah oleh magnet. Satuan intensitas magnet menurut sistem metrik pada Satuan Internasional (SI) adalah Tesla dan SI unit untuk total fluks magnetik adalah weber.  $1 \text{ weber/m}^2 = 1 \text{ tesla}$ , yang memengaruhi satu meter persegi.[5]

Jenis-jenis magnet dapat dibedakan menjadi 3 bagian yaitu magnet tetap, magnet tidak tetap dan magnet buatan.

### 1. Magnet tetap

Magnet tetap tidak memerlukan tenaga atau bantuan dari luar untuk menghasilkan daya magnet (berelektromagnetik). Jenis magnet tetap selama ini yang diketahui terdapat pada:

- Magnet neodmium, merupakan magnet tetap yang paling kuat. Magnet neodmium (juga dikenal sebagai NdFeB, NIB, atau magnet Neo),

merupakan sejenis magnet tanah jarang, terbuat dari campuran logam neodymium, magnet inilah yang digunakan untuk alat ini.

- Magnet Samarium-Cobalt: salah satu dari dua jenis magnet bumi yang langka, merupakan magnet permanen yang kuat yang terbuat dari paduan samarium dan kobalt.
- Ceramic Magnets
- Plastic Magnets
- Alnico Magnets

## 2. Magnet tidak tetap

Magnet tidak tetap tergantung pada medan listrik untuk menghasilkan medan magnet. Contoh magnet tidak tetap adalah elektromagnet.

## 3. Magnet buatan

Magnet buatan meliputi hampir seluruh magnet yang ada sekarang ini.

Bentuk magnet buatan antara lain:

- Magnet U
- Magnet ladam
- Magnet batang
- Magnet lingkaran
- Magnet jarum (kompas)

## 2.6 Teori Kontrol Proporsional Integral Differensial ( PID )

Sistem pengendali merupakan suatu sistem yang difungsikan untuk mengendalikan suatu sistem yang lain. Sistem pengendali digunakan agar kinerja suatu sistem kendali menjadi lebih baik atau pasti. Secara umum sistem pengendalian terbagi menjadi dua jenis yaitu *Open Loop Control System* dan *Closed Loop Control System* <sup>[6]</sup>. Pada sistem pengendali dikenal beberapa istilah, antara lain SP, error, MV, PV, dan Plant, yaitu adalah:

- SP (*Set Point*) adalah harga atau nilai dari keadaan yang ingin dicapai pada proses.
- Error adalah selisih antara *Set Point* dan *Process Variable*.



- MV (*Manipulated Variable*) adalah harga atau nilai yang diatur agar proses menjadi stabil. *Manipulated Variable* biasanya dihubungkan dengan input aktuator (contoh: *control valve*).
- PV (*Process Variable*) adalah sinyal hasil pemantauan terhadap proses atau *plant*. *Process Variable* umumnya adalah hasil pembacaan dari suatu sensor (contoh: *thermocouple*).
- *Plant* adalah objek yang akan dikendalikan (contoh: temperatur).

*Open Loop Control System* atau sistem pengendali *loop* terbuka merupakan sistem pengendalian dimana objek yang dikontrol tidak di-*feedback* ke pengendali, sehingga pengendali hanya akan memberikan output jika diberikan suatu sinyal input. Pengendali jenis ini masih bersifat manual karena tidak akan terlepas dari intervensi atau campur tangan manusia. Pengendali ini tidak akan bekerja secara otomatis, karena masih adanya intervensi manusia dan hasil dari suatu proses yang dikendalikan tidak dibandingkan oleh pengendali itu sendiri. Gambar 2.5 menggambarkan sistem pengendali *loop* terbuka (*Open Loop Control System*).



Gambar 2.12. Sistem Pengendali *Loop* Terbuka<sup>[6]</sup>

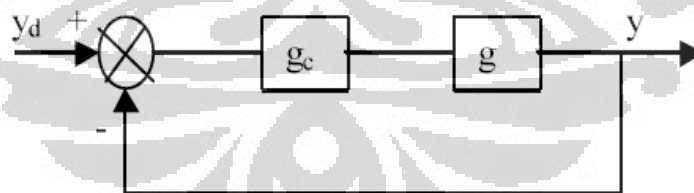
Sistem pengendali yang kedua adalah *Closed Loop Control System* atau sistem pengendali *loop* tertutup, yaitu sistem pengendalian dimana objek yang dikontrol di-*feedback* ke input pengendali. Input yang diberikan ke pengendali merupakan selisih antara besaran (PV) dan besaran (SP). Nilai selisih ini sering disebut dengan *error*. Tujuan dari pengendali adalah membuat nilai *Process Variable* (PV) sama dengan nilai *Set Point* (SP), atau nilai *error* = 0. Sinyal *error* akan diolah oleh pengendali agar nilai (PV) sama dengan nilai (SP). Pengendali jenis ini bersifat otomatis karena objek yang akan dikendalikan dibandingkan lagi dengan input keadaan yang diinginkan, sehingga intervensi manusia dapat dihilangkan. Kinerja dari suatu pengendali ditentukan oleh semakin cepatnya

respon pengendali untuk mengubah MV terhadap perubahan sinyal error, dan semakin kecilnya kesalahan yang terjadi. Gambar 2.13 menggambarkan sistem pengendali *loop* tertutup (*Closed Loop Control System*).



Gambar 2.13. Sistem Pengendali *Loop* Tertutup<sup>[6]</sup>

Seiring dengan berkembangnya penelitian tentang identifikasi suatu sistem “*black box*”, maka memperoleh transfer function atau karakteristik dari sistem tersebut bukanlah hal yang teramat sulit. Hal ini menyebabkan metode tuning kontroler yang membutuhkan model plant sebenarnya juga dapat dilakukan dengan relatif mudah, misalnya dengan metode *Direct Synthesis*. Metode ini terlebih dulu menentukan perilaku output yang diinginkan (*reference*) dengan membuat bentuk trayektorinya, dan model prosesnya (*plant*) digunakan untuk secara langsung mendapatkan persamaan kontroler yang sesuai. Berikut ini penurunan rumusnya [7]. Jika diketahui diagram blok dari suatu system ialah sebagai berikut.



Gambar 2.14. Blok diagram dari *system control close-loop* [7]

Maka *closed-loop* transfer function ialah sebagai berikut :

$$y(s) = \frac{g_p g_c}{1 + g_p g_c h} y_d(s) \quad (2.3)$$

Dan pendekatan yang diinginkan untuk mendapatkan *set point* yang baru dimodelkan dengan trayektori yang diinginkan berikut :

$$\frac{y(s)}{y_d(s)} = q(s) = \frac{g g_c}{1 + g g_c h} \quad (2.4)$$

sehingga persamaan kontrolernya :

$$g_c = \frac{1}{g} \left( \frac{q}{1-q} \right) \quad (2.5)$$

Sesuai dengan transfer function plant motor DC yang telah didapatkan dan berbentuk *First Order Plus Dead Time* (FOPDT), yaitu :

$$g(s) = \frac{K \cdot e^{-\alpha s}}{\tau_s + 1} \quad (2.6)$$

Dengan memasukkan hasil Persamaan 2.4 dan Persamaan 2.5 pada Persamaan 2.6 maka didapat persamaan kontroler :

$$g_c = \frac{(\tau_s + 1)}{K} \left( \frac{1}{\tau_r s + 1 - e^{-\alpha s}} \right) \quad (2.7)$$

Akhirnya didapatkan kontroler dalam bentuk persamaan, namun untuk merealisasikannya sangat sulit karena besaran  $e^{-\alpha s}$  tidak bisa diimplementasikan dalam komponen analog. Namun dengan adanya implementasi kontroler PID pada mikroprosesor dan komputer digital membuat besaran tersebut bisa diimplementasikan. Melalui model dasar kontroler ini didapatkan beberapa macam nilai tuning PID yang berbeda – beda.

Dengan menggunakan pendekatan Pade orde 1 :

$$e^{-\alpha s} \approx \frac{1 - \frac{\alpha}{2}s}{1 + \frac{\alpha}{2}s} \quad (2.8)$$

Pada Persamaan 2.8, kontroler yang didapatkan menjadi

$$g_c = \frac{\tau}{K(\tau_r + \alpha)} \left( 1 + \frac{1}{\tau s} \right) \left( \frac{1 + \frac{\alpha}{2}s}{1 + \tau^* s} \right) \quad (2.9)$$

dengan  $\tau^*$  adalah filter yang mempunyai persamaan sebagai berikut:

$$\tau^* = \frac{\alpha \tau_r}{2(\alpha + \tau_r)} \quad (2.10)$$

Persamaan 2.8 mempunyai struktur sesuai dengan struktur kontroler PID komersial. Maka parameter kontroler PID komersial dapat dicari sebagai berikut :

$$K_c = \frac{\tau}{K(\tau_r + \alpha)}; \tau_I = \tau; \tau_D = \frac{\alpha}{2}; \tau^* = \frac{\alpha}{2} \left( \frac{\tau_r}{\alpha + \tau_r} \right) \quad (2.11)$$

Dengan kontroler yang sama, persamaan di atas dapat disusun kembali menjadi:

$$g_c = \frac{\tau + \frac{\alpha}{2}}{K(\alpha + \tau_r)} \left[ 1 + \frac{1}{\left( \tau + \frac{\alpha}{2} \right) s} + \left( \frac{\alpha \tau}{\tau + \frac{\alpha}{2}} \right) \left( \frac{1}{\tau^* s + 1} \right) \right] \quad (2.12)$$

Maka parameter tuning dari kontroler PID ideal ialah :

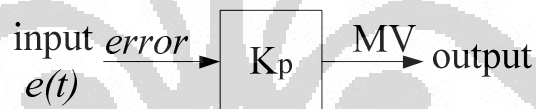
$$K_c = \frac{\tau + \frac{\alpha}{2}}{K(\tau_r + a)}; \tau_I = \tau + \frac{\alpha}{2}; \tau_D = \frac{\frac{\alpha}{2}\tau}{\tau + \frac{\alpha}{2}}; \tau^* = \frac{\alpha}{2} \left( \frac{\tau_r}{\alpha + \tau} \right) \quad (2.13)$$

Parameter inilah yang digunakan dalam eksperimen.

Pengendali *P.I.D* terdiri dari tiga macam pengendali yaitu pengendali *Proportional* (P), pengendali *Integral* (I) dan pengendali *Differensial* (D). Masing-masing pengendali ini saling dikombinasikan sehingga didapatkan bentuk atau struktur dari *P.I.D*, yaitu struktur paralel atau struktur mix. Berikut ini adalah penjelasan dari masing-masing pengendali.

### 2.6.1 Pengendali *Proportional* (P)

Pengendali *proportional* berfungsi untuk mengalikan sinyal input dengan suatu besaran atau konstanta dengan nilai tertentu.

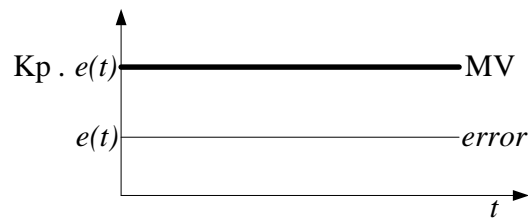


Gambar 2.15. Blok Diagram Pengendali *Proportional*<sup>[6]</sup>

Persamaan hubungan antara input (*error*) dan output (MV) pada pengendali ini adalah

$$\begin{aligned} \text{output} &= K_p \cdot \text{input} \\ \text{MV} &= K_p \cdot e(t) \end{aligned} \quad (2.14)$$

Karena pengendali *proportional* hanya menguatkan sinyal input saja, maka hubungan antara sinyal *error* dan sinyal MV dapat digambarkan seperti grafik respon berikut ini.



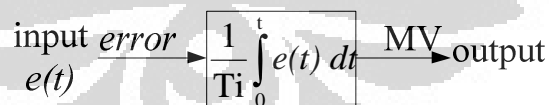
Gambar 2.16. Grafik respon Pengendali *Proportional*<sup>[6]</sup>

Pengendali *proportional* berfungsi untuk mempercepat proses yang dikendalikan menuju ke keadaan *set-point*. Kecepatan proses ini sangat bergantung dari besarnya nilai  $K_p$  pada pengendali *proportional*.

Semakin besar nilai  $K_p$  maka semakin besar juga penguatannya sehingga respon dari pengendali akan semakin cepat juga dan akan mengurangi besarnya *steady-state error*. Tetapi jika nilai  $K_p$  terlalu besar maka sistem akan mengalami *over shoot* yang besar sehingga proses yang dikendalikan menjadi tidak stabil bahkan akan mengalami osilasi.

### 2.6.2 Pengendali *Integral* (I)

Pengendali *integral* berfungsi untuk meng-*integral*-kan sinyal input lalu dibagi dengan suatu besaran atau konstanta dengan nilai tertentu.

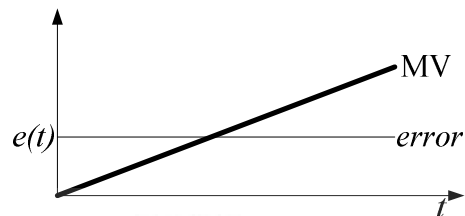


Gambar 2.17. Blok Diagram Pengendali *Integral*<sup>[6]</sup>

Persamaan hubungan antara input (*error*) dan output (MV) pada pengendali ini adalah

$$MV = \frac{1}{T_i} \int_0^t e(t) dt \quad (2.15)$$

Karena pengendali *integral* hanya meng-*integral*-kan sinyal input saja, maka hubungan antara sinyal *error* dan sinyal MV dapat digambarkan seperti grafik respon berikut ini.



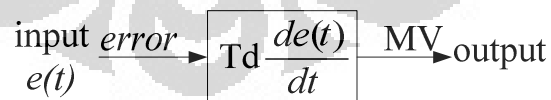
Gambar 2.18. Grafik Respon Pengendali *Integral*<sup>[6]</sup>

Pengendali *integral* berfungsi untuk mengurangi dan menghilangkan *steady-state error* yang timbul setelah respon *plant* dari pengendali *proportional* sudah stabil. Semakin kecil nilai *steady-state error*, maka respon dari *plant* akan semakin mendekati keadaan *steady-state*. Semakin kecil nilai *error* maka semakin kecil juga nilai *timing integral*-nya, sehingga kurva MV akan semakin landai.

Pengendali *integral* sangat optimal bekerja pada daerah di sekitar titik *set-point*, yaitu antara *steady-state error* dan *set point*.

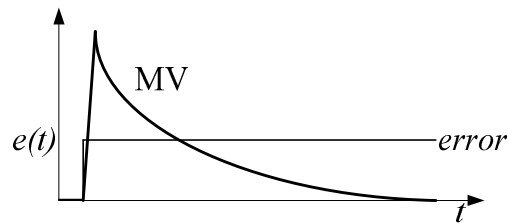
### 2.6.3 Pengendali *Diferensial* (D)

Pengendali *diferensial* berfungsi untuk men-*diferensial*-kan sinyal input lalu dikalikan dengan suatu besaran atau konstanta dengan nilai tertentu.



Gambar 2.19. Blok Diagram Pengendali *Diferensial*<sup>[6]</sup>

Persamaan hubungan antara input (*error*) dan output (MV) pada pengendali ini adalah karena pengendali *diferensial* hanya meng-*diferensial*-kan sinyal input saja, maka hubungan antara sinyal *error* dan sinyal MV dapat digambarkan seperti grafik respon berikut ini.



Gambar 2.20. Grafik Respon Pengendali *Differensial*<sup>[6]</sup>

Pengendali *differensial* berfungsi untuk mengurangi respon yang terlalu berlebih yang dapat mengakibatkan *over shoot* pada proses *plant* karena nilai  $K_p$  yang terlalu besar pada pengendali *proportional*. Output dari pengendali *differensial* akan bernilai sangat besar jika perubahan *error* sangat besar. Perubahan *error* yang sangat besar ini terjadi ketika proses *plant* bergerak menuju ke titik *set-point* dalam waktu yang sangat singkat (nilai  $dt$  sangat kecil). Hal ini disebabkan karena respon pengendali yang terlalu cepat akibat terlalu besarnya nilai  $K_p$  pada pengendali *proportional*. Pengendali *differensial* hanya akan bekerja ketika terjadi perubahan *error*, sehingga ketika proses yang dikendalikan sudah stabil maka pengendali *differensial* sudah tidak bekerja lagi.

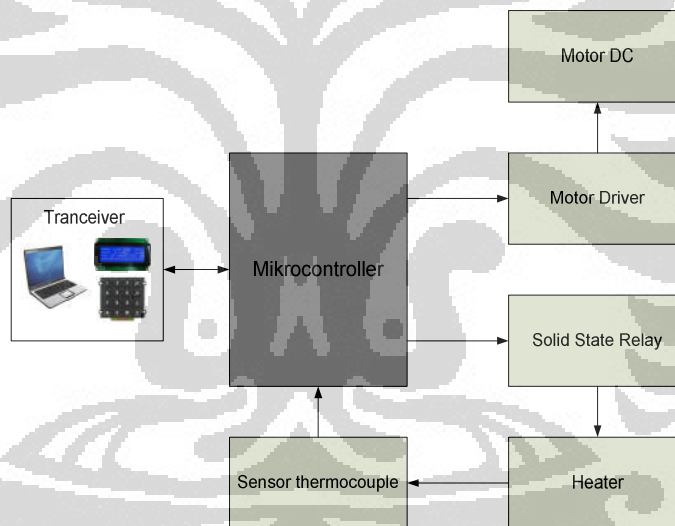


## BAB 3 PERANCANGAN DAN CARA KERJA SISTEM

Pada bab ini akan dibahas mengenai perancangan sistem beserta cara kerja dari masing-masing konstruksi mekanik, hardware, dan Software yang digunakan penulis dalam penyusunan alat ” Hotplate stirrer magnetik terkendali temperatur dan kecepatan pengaduk”.

### 3.1 Sistem Kerja Alat

Alat ini dirancang agar dapat mengendalikan temperatur dan kecepatan pengaduk sesuai dengan yang diinginkan dan memiliki sistem pemanas. Berikut ini adalah gambaran secara umum dari alat ini:



Gambar 3.1. Blok diagram cara kerja alat

Dari diagram blok di atas keypad sebagai input untuk menginisialisasikan *microcontroller* yang bertindak sebagai pengendali, dimana sebagai tranciever yang mengirimkan nilai *Set Point (SP)* yang berupa bilangan dan menampilkan nilai *Process Variable (PV)*. Prinsipnya pengendali pada alat ini yaitu kecepatan motor dan temperatur. Dalam pengendalian Stirrer, nilai dari *Set Point* berupa bilangan akan masuk ke PWM (*Pulse Width Modulation*) sebagai output PWM yang akan mengatur kecepatan motor DC. Pada pengendalian temperatur

dikendalikan dengan proses PID, Dari temperatur heater tersebut dibaca dengan *sensor thermocouple* yang akan menghasilkan nilai temperatur ( $^{\circ}\text{C}$ ) dan nantinya akan kembali ke dalam sistem pengendali berupa bilangan sebagai nilai *Manipulated Variable* (MV). Untuk bagian pengendali, PWM dan *counter* terdapat di dalam *microcontroller*. Dalam pengendali temperatur. Sistem pengendali yang digunakan untuk pengendali temperaturnya adalah *PI controller* (*Proportional Integrator*).

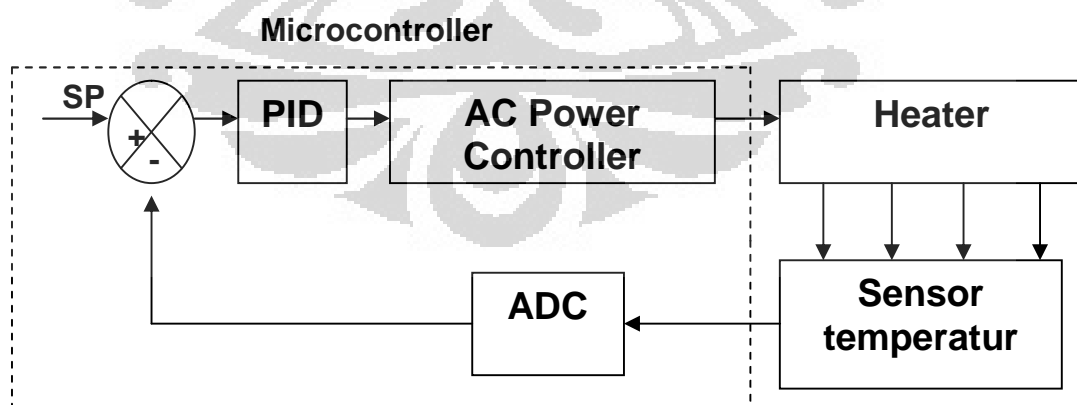
Di dalam pengendali temperatur (Gambar 3.2), ditentukan nilai *set point* berupa bilangan lalu dihitung nilai *error*-nya dengan persamaan :

$$E = SP - PV \quad (3.1)$$

Dimana SP adalah Set Point dan PV adalah *Process Variable*, lalu ke dalam PID yang akan dihitung nilai *Manipulated Variable* (MV) dengan persamaan:

$$MV = K_p \left( E + K_i \int E dt + K_d \frac{dE}{dt} \right) \quad (3.2)$$

Nilai MV tersebut akan mengatur keluaran *AC power controller* untuk mengatur sistem pemanasnya yaitu heater dengan demikian reaktor mengalami pemanasan.



Gambar 3. 2. Blok diagram pengendali temperatur

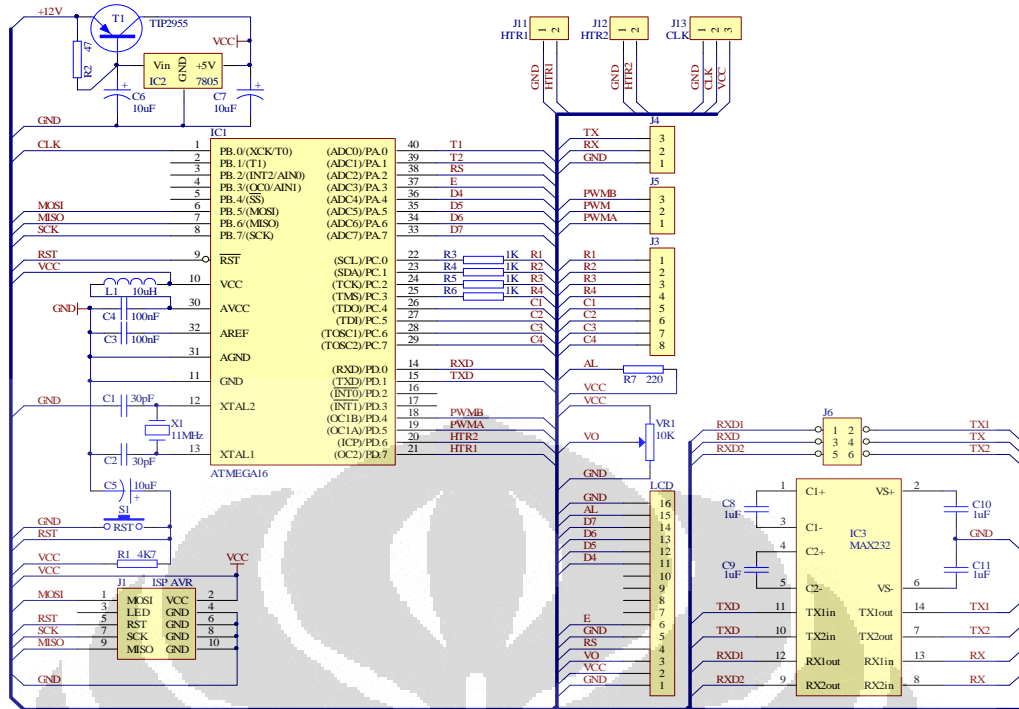
Di dalam heater tersebut terdapat sensor temperatur, karena temperatur yang diharapkan dapat mengukur temperatur hingga 250°C maka menggunakan sensor temperatur termokopel tipe K, karena sensor temperatur ini untuk rentang suhu -200°C hingga 1200°C dan selain itu termokopel tipe K ini harganya lebih murah. Keluaran dari sensor tersebut akan dimasukkan ke dalam ADC *microcontroller* 10 bit dan akhirnya dihasilkan nilai PV (*Process Variable*) yang nantinya akan kembali dihitung nilai error dari sistem tersebut.

### 3.2 Perancangan Perangkat Keras (Hardware)

Di dalam pembuatan alat ini perangkat kerasnya (hardware) terdapat beberapa bagian, yakni rangkaian *minimum system*, rangkaian *driver* motor, rangkaian *cold junction*, rangkaian *keypad*.

#### 3.2.1 Rangkaian Minimum Sistem ATmega16

Rangkaian minimum sistem yang digunakan penulis pada tugas akhir ini yaitu menggunakan mikrokontroler AVR tipe ATmega 16. Mikrokontroler adalah suatu piranti yang digunakan untuk mengolah data-data biner (digital) yang didalamnya merupakan gabungan dari rangkaian-rangkaian elektronik yang dikemas dalam bentuk suatu *chip* (IC). Pada umumnya mikrokontroler terdiri dari bagian-bagian sebagai berikut: Alamat (*address*), Data, Pengendali, Memori (*RAM* atau *ROM*), dan bagian *input-Output*. AVR merupakan seri mikrokontroler CMOS 8-bit buatan Atmel, berbasis arsitektur RISC (Reduced Instruction Set Computer). Hampir semua instruksi dieksekusi dalam satu siklus clock. AVR mempunyai 32 register general-purpose, timer/counter fleksibel dengan mode compare, interrupt internal dan eksternal, serial UART, programmable Watchdog Timer, dan mode power saving. Mempunyai ADC dan PWM internal.



Gambar 3.3. Rangkaian minimum sistem ATmega16

Rangkaian minimum sistem ini terdiri dari rangkaian mikrokontroler dan ISP programmer, kristal, 4 buah header yang terhubung ke port I/O-nya mikrokontroler. Sebagai otak dari sistem ini menggunakan mikrokontroler produk dari atmel keluarga AVR seri ATmega 16. Rangkaian ini dibutuhkan agar dapat mengisi (me-download) program yang telah dibuat. Program tersebut meliputi program pengendalian suhu dan kecepatan serta pengiriman data komunikasi serial. Port yang digunakan pada *microcontroller* Atmega16 adalah PA untuk LCD, PC untuk keypad dan PD untuk komunikasi, motor dan *heater* (hot plate).

### 3.2.2 Konsep I/O pada mikrokontroler AVR ATmega16

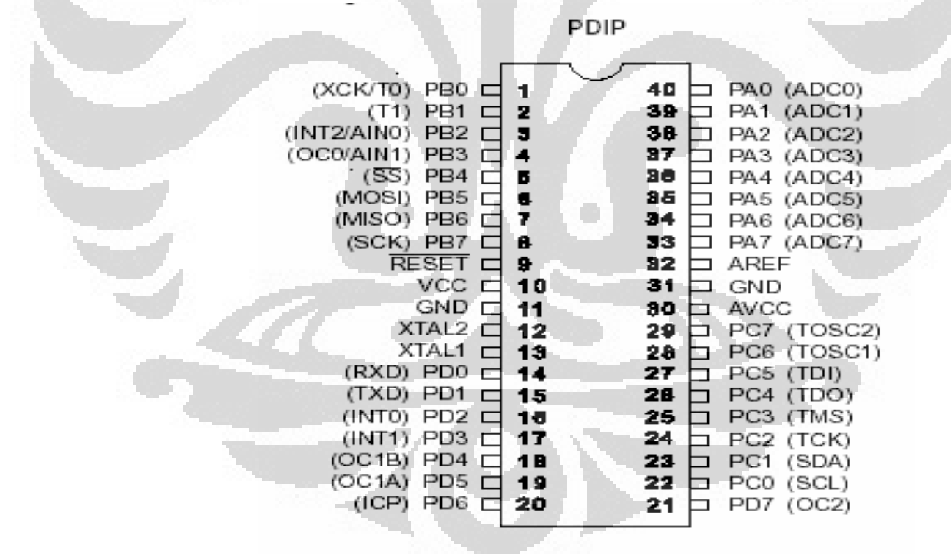
Pemrograman I/O pada mikrolontroller merupakan dasar dari prinsip pengontrolan berbasis mikrokontroler, dimana orientasi dari penerapan mikrokontroler ialah untuk mengendalikan suatu system berdasarkan informasi input yang di terima, lalu diproses oleh mikrokontroler dan dilakukan aksi pada bagian output sesuai program yang telah di tentukan sebelumnya.pada gambar diatas, terdapat empat buah port, yaitu PA,PB,PC, dan PD yang semuanya dapat deprogram sebagai input ataupun output.Jika dilihat lebih detail lagi pada

pemroses mikrokontroller ini , terdapat unit CPU utama untuk memastikan eksekusi program.CPU juga dapat mengakses memori, melakukan kalkulasi,pengontrolan dan penanganan interupsi dengan menggunakan arsitektur Harvard ( bus untuk memori dan program dan data terpisah ) sehingga di hasilkan performa yang tinggi (Budiharto, Widodo ,hal 40).

Hal ini di karenakan instruksi pada memori program di eksekusi dengan single level pipelining, dengan demikian pada saat sebuah instruksi dieksekusi, instruksi berikutnya dapat diskses dari memori program.

### 3.2.3 Konfigurasi Pin

Pin-pin pada ATmega16 dengan kemasan 40-pin DIP (dual inlinepackage) ditunjukkan oleh gambar3.4. Kemasan pin tersebut terdiri dari 4 Port yaitu Port A, Port B, Port C,Port D yang masing-masing Port terdiri dari 8 buah pin. Selain itu juga terdapat RESET, VCC, GND 2 buah, VCC, AVCC, XTAL1, XTAL2 dan AREF.



Gambar 3.4. Pin-pin ATmega16 kemasan 40-pin

Diskripsi dari pin-pin ATmega 16L adalah sebagai berikut :

1. **VCC** : Supply tegangan digital.
2. **GND** : Ground
3. **PORT A** : Merupakan pin I/O dua arah dan dapat diprogram sebagai pin masukan ADC.

4. **PORT B** : Merupakan pin I/O dua arah dan pin fungsi khusus, yaitu timer/counter, komparator analog, dan ISP.
5. **PORT C** : Merupakan pin I/O dua arah dan pin fungsi khusus, yaitu TWI, komparator analog, dan timer osilator. Pin port C adalah tri-states ketika kondisi sebuah reset menjadi aktif, sekalipun clocknya tidak jalan. Jika interface JTAG enable, pull up resistor di pin PC5(TDI), PC3(TMS), dan PC2(TCK) akan aktif sekalipun reset terjadi.
6. **PORT D** : Merupakan pin I/O dua arah dan pin fungsi khusus, yaitu komparator analog, interupsi eksternal, dan komunikasi serial.
7. **RESET** : Pin yang digunakan untuk mereset mikrokontroler. Sebuah low level pada pin akan lebih lama dari pada lebar pulsa minimum akan menghasilkan reset meskipun clock tidak berjalan.
8. **XTAL1** : Input inverting penguat Oscilator dan input intenal clock operasi rangkaian.
9. **XTAL2** : Output dari inverting penguat Oscilator.
10. **AVCC** : Pin supply tegangan untuk Port A dan A/D converter. Sebaiknya eksternalnya dihubungkan ke VCC meskipun ADC tidak digunakan. Jika ADC digunakan seharusnya dihubungkan ke VCC melalui low pas filter.
11. **AREF** : Pin referensi analog untuk A/D konverter.

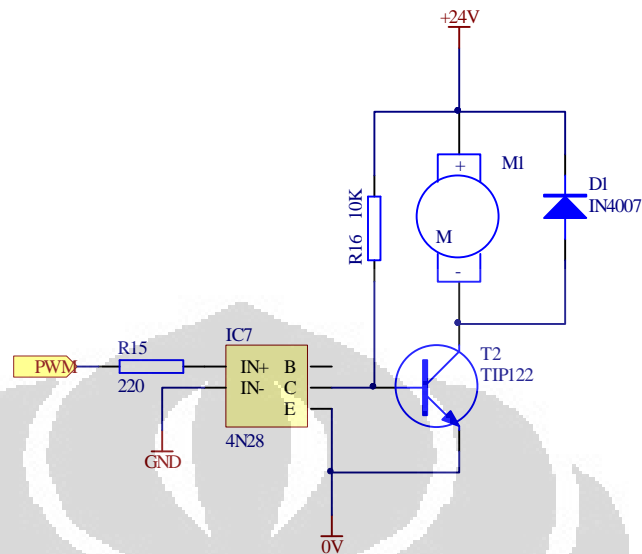
#### 3.2.4 Rangkaian Driver Motor

Sebuah motor DC dapat dipercepat putarannya dengan menambahkan nilai tegangan yang melalui kumparan kawat pada motor DC. Pada sistem elektronik ini dapat dibuat suatu rangkaian pengendali yang dapat mempercepat putaran motor secara otomatis dengan cara mengatur lebar dari pulsa (*Pulse Width Modulation*), karena didalam motor DC juga terdapat sensor. Transistor pada rangkaian pengendali motor DC ini digunakan sebagai saklar elektronik

Rangkaian pada Gambar 3.5 merupakan rangkaian pengendali yang dapat mengendalikan kecepatan putaran sebuah motor DC.

Adapun cara kerja rangkaian motor driver ini adalah jika pada PWM diberi logika 0 ( atau diberi tegangan 0 Volt ), maka IC 4N28 ( sebagai saklar ) akan non

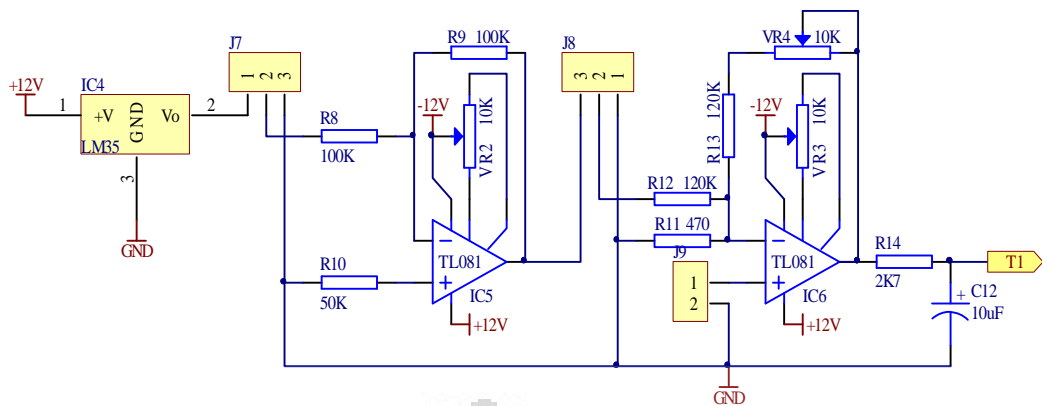
aktif sehingga akan memberikan nilai pada M sesuai sebesar 24 Volt DC dan motor akan aktif.



Gambar 3.5. Rangkaian driver motor

### 3.2.5 Rangkaian Cold Junction

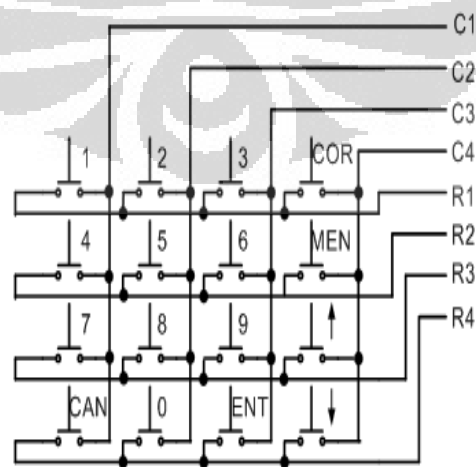
*Thermocouple* yang digunakan pada alat ini dihubungkan dengan rangkaian *cold junction* kemudian dikalibrasikan terlebih dahulu. Memutuskan hubungan antara rangkaian TL081 dan rangkaian LM35, lalu mengukur tegangan output yang disebut tegangan offset. Mengatur tegangan offset menjadi 0. Keluaran pada rangkaian LM35 adalah 300mV bila suhu ruangan adalah 30°C karena besar temperatur akan sama dengan besar tegangan, dengan kenaikan 10mV/°C. *Thermocouple* dihubungkan ke rangkaian dan dicelupkan ke dalam air mendidih. Mengatur potensio agar keluaran merupakan selisih dari temperatur suhu ruangan dan temperatur air mendidih yaitu 700mV. Rangkaian LM35 lalu dihubungkan dengan rangkaian *thermocouple* dan keluarannya adalah 1V (300mV ditambah 700mV).



Gambar 3.6. Rangkaian *cold junction*

### 3.2.6 Rangkaian Keypad

*Keypad* sering digunakan sebagai suatu input pada beberapa peralatan yang berbasis mikroprosesor atau mikrokontroler. *Keypad* sesungguhnya terdiri dari sejumlah saklar, yang terhubung sebagai baris dan kolom dengan susunan seperti yang ditunjukkan pada gambar. Agar mikrokontroler dapat melakukan scan keypad, maka port mengeluarkan salah satu bit dari 4 bit yang terhubung pada kolom dengan logika low “0” dan selanjutnya membaca 4 bit pada baris untuk menguji jika ada tombol yang ditekan pada kolom tersebut. Sebagai konsekuensi, selama tidak ada tombol yang ditekan, maka mikrokontroler akan melihat sebagai logika high “1” pada setiap pin yang terhubung ke baris.



Gambar 3.7. Rangkaian dasar keypad



### 3.3 Perancang Perangkat Lunak (*software*)

Pada alat ini menggunakan perangkat lunak (*software*) didalam pengendaliannya. Software yang digunakan adalah Bascom AVR dan pengambilan data secara manual dengan melihat data langsung pada display LCD atau komputer menggunakan LabView. Setelah mendapatkan model dan mengetahui fungsi transfers dari data temperatur dengan menggunakan metode *Direct Synthesis* maka dibuat program Bascom AVR untuk mengendalikan sistem tersebut.

#### 3.3.1 Pengambilan Data

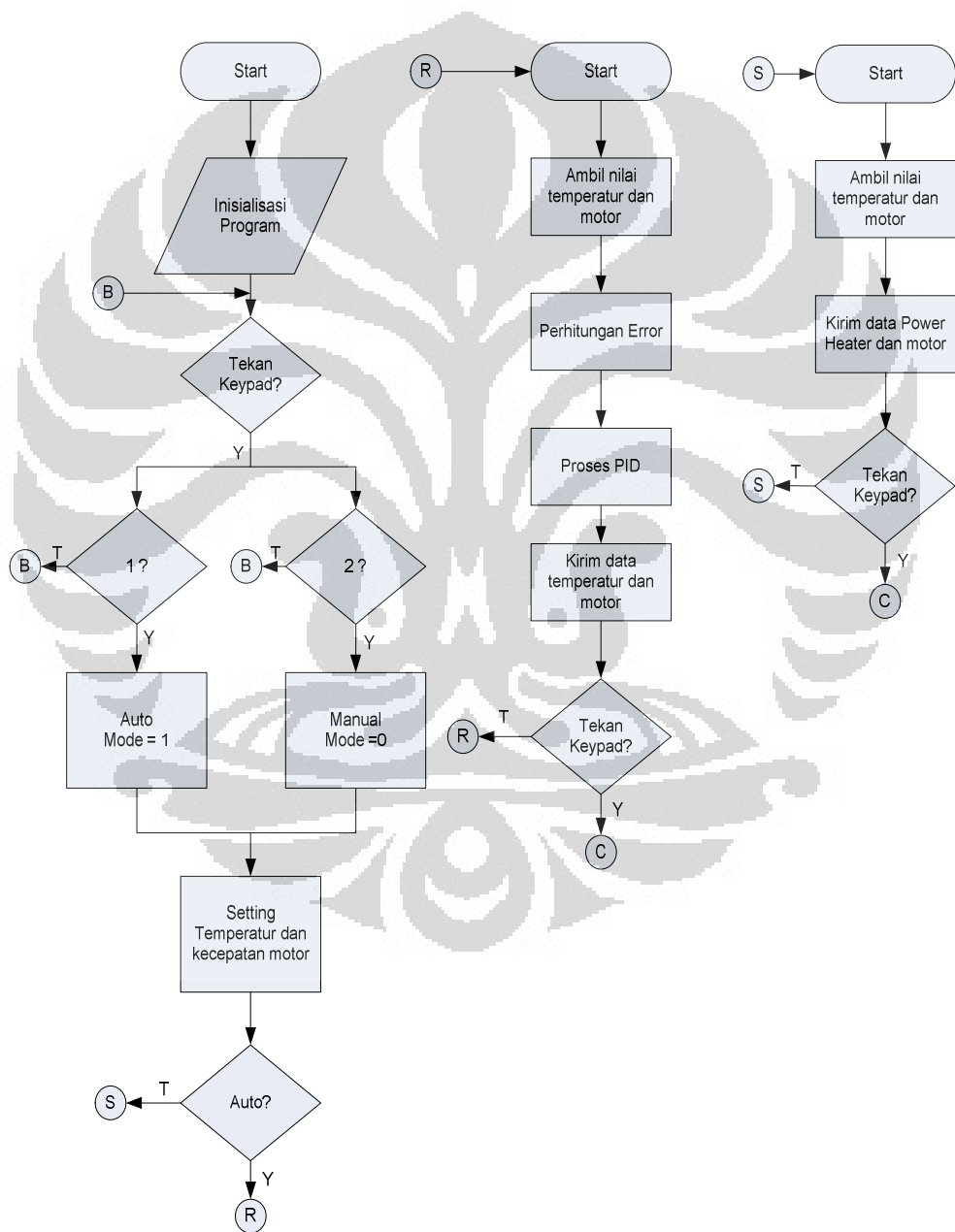
Ada tiga proses untuk mengambil data, yaitu untuk mengambil data temperatur, power motor dan waktu. Bila salah satu dari tombol perintah ditekan maka *lowerline* pada LCD akan berubah sesuai dengan tombol yang ditekan. Kemudian memasukkan nilai temperatur, power motor, dan waktu yang diinginkan dan sistem akan menunggu hingga tombol \* ditekan. Setelah tombol \* ditekan maka nilai tersebut tersimpan dalam eeprom dan sistem balik ke awal program.

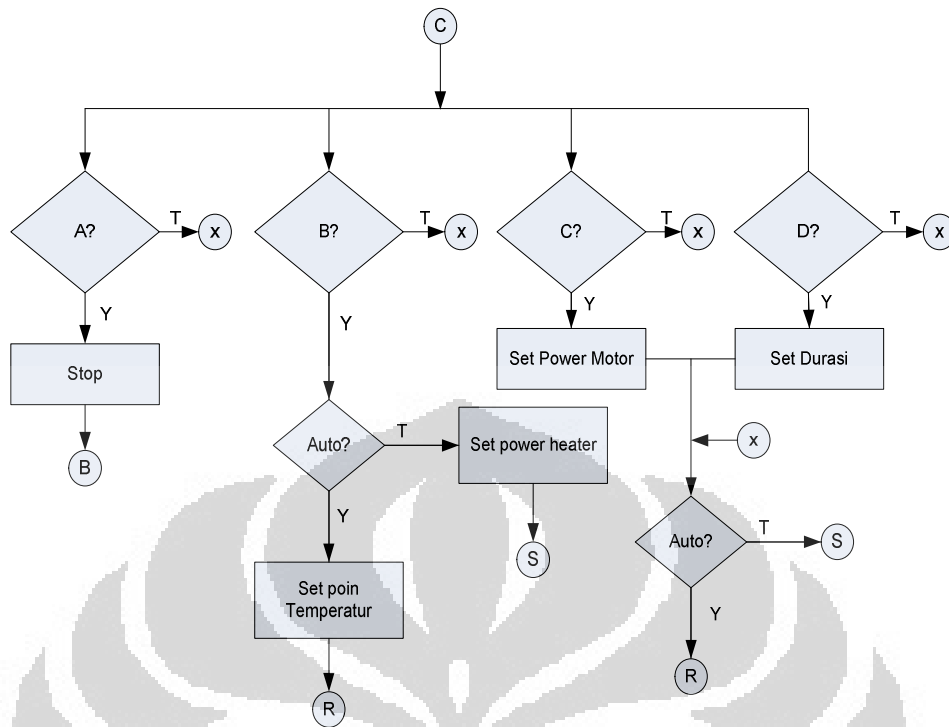
#### 3.3.2 Proses Pada Sistem Minimum Atmega16

Pada flowchart dibawah adalah pengendali untuk heater dimana pada Atmega16 ini inputnya dari keypad. Awalnya Atmega16 ini menginisialisasi perintah dari keypad yaitu mengisi data yang akan dimasukkan kedalam rumus untuk dijalankan. Apabila temperatur, power motor, dan Time sudah disimpan pada eeprom, maka data-data tersebut akan diolah untuk diproses.  $K_p$ ,  $T_i$ , dan  $T_d$  yang telah diatur akan diolah AVR kedalam rumus bersamaan PID, sedangkan data-data temperatur, power motor, dan Time akan langsung dibandingkan untuk dijalankan karena didalam program AVR telah terdapat persamaan yang telah diambil dari pengambilan dari data alat sebelumnya. Data data pada eeprom adalah data input dari keypad.

Apabila sudah membaca data pada eeprom AVR akan membaca nilai SP yaitu nilai untuk mengendalikan alat. Nilai SP ini juga inputan dari keypad setelah membaca nilai SP kemudian di Start maka AVR akan memproses data-data tersebut hingga nilai dari semua data terpenuhi. Proses ini akan berjalan terus -

menerus sampai program di STOP, atau waktu telah habis. Pengendalian temperature dilakukan dengan mengendalikan tegangan yang lewat pada Solid State Relay (SSR), sehingga SSR akan otomatis nyala mati untuk menstabilkan tegangan untuk heater sesuai dengan perintah pada persamaan PID yang ada pada mikrokontroler. Akan tetapi untuk motor, pengendalian dilakukan dengan mengatur pulsa yang terdapat pada motor DC yang menggunakan PWM. Berikut adalah Flow chart program :





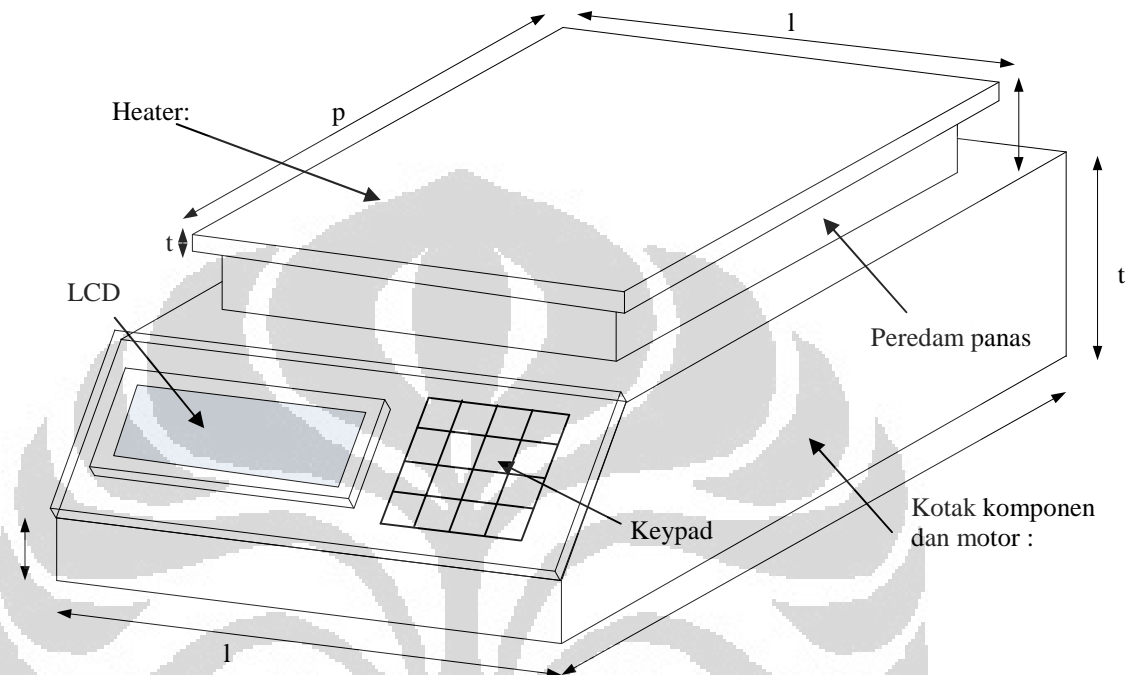
Gambar 3.8. *Flowchart* program pengendali

Dari flowchart diatas dapat penulis jelaskan bahwa ketika start terjadi inialisasi LCD diikuti dengan ON- nya mikro. Kemudian cek keypad. Ketika input keypad = 1, maka auto\_mode = 1, program auto akan aktif. Namun ketika keypad = 2 maka auto\_mode = 0, program manual akan aktif.

Kemudian setting set point temperatur, kecepatan motor dan durasi. Setelah masuk dalam program utama Fungsi dari keypad A sebagai stop dan kembali ke program awal, keypad B sebagai sebagai set point temperatur atau power heater. keypad C sebagai sebagai set point kecepatan stirrer atau power motor, dan keypad D sebagai setting waktu (durasi).

### 3.4 Perancangan Mekanik

Pada bagian ini di ada 3 bagian yaitu heater yang terdiri dari pemanas dan thermocouple, pemisah (peredam panas), dan kotak komponen.



Gambar 3.9. Konsep pembuatan mekanik

#### 3.4.1 Perancangan Heater

Dalam perancangan heater ini kita menggunakan heater tubular. heater tubular ini mampu menghasilkan panas hingga  $350^{\circ}\text{C}$  yang dibentuk lingkaran, kemudian di letakan pada plate yang terbuat dari stainless steel. Plate stainless ini dinamakan hot plate yaitu berupa plate persegi. Penggunaan heater ini menggunakan daya 500 watt. Diameter hot plate tersebut adalah 15cm dan hot plate ini bentuknya persegi maka ketebalannya yaitu 0,5cm. Ukuran secara keseluruhan bodi hotplate ini adalah 18cmx16cm dengan ketebalan 1,5cm.

Spesifikasi Heater

|                   |         |
|-------------------|---------|
| Tebal plate (t)   | = 1,5cm |
| Lebar plate (l)   | = 16cm  |
| Panjang plate (p) | = 18cm  |

(a) Heater tubular sebelum digabung



(b) Heater setelah di gabung dengan plate



Gambar 3.10. Perancangan heater

### 3.4.2 Perancangan Kotak komponen dan motor DC

Dalam perancangan kotak ini perlu di pastikan terlebih dahulu tinggi ukuran motor DC dan besarnya rangkaian elektronika. kotak ini menggunakan bahan stainless steel karena bahan tersebut dapat menahan panas yang cukup tinggi yaitu kira-kira sampai 1000 °C. Selain itu stainless steel juga anti karat sehingga kita tidak perlu khawatir akan rusaknya mekanik tersebut. Ukuran kotak ini dengan tinggi 8 cm, lebar 18cm dan panjang 22cm. Dalam kotak ini terdapat rangkaian driver motor, minsis, cold junction, power supply, beserta motor DC . dibagian depan ada sebuah LCD dan keypad dan bagian belakang untuk soket supply, saklar, fuse, dan konektor DB9 (untuk komunikasi ke computer) . Bagian

atas di lubangi untuk motor dan tiang heater. Bagian lubang tersebut di beri gypsum agar uap panas tak masuk kedalam ruang kotak.

Penggunaan motor berfungsi untuk penggerak magnet pengaduk yang akan dicoba nantinya untuk larutan kimia. Pengaduk yang dipakai menggunakan bahan magnet yang dilapiskan teflon. Penggunaan bahan ini dimaksudkan karena bahan teflon tidak berpengaruh pada pencampuran bahan kimia yang akan diuji nantinya, dimana anti lengket (berisi zat abheret yaitu dapat mencegah melekatnya benda-benda) teflon secara kimia tidak akan berikatan dengan apa pun. Ukuran magnet pengaduk disesuaikan dengan diameter gelas kimia.



Gambar 3.11. Perancangan Kotak

### 3.4.3 Perancangan Peredam Panas

Pada kotak komponen terdapat komponen dan rangkaian yang rentan terhadap temperature tinggi. Sehingga perlu adanya pereredam panas agar perembetan panas yang dihasilkan oleh hotplate tidak merambat ke kotak komponen. Maka dirancang Peredam panas ini dengan menggunakan gypsum (kapur) yang berfungsi untuk meredam panas dan pemisah penyangga hotplate sehingga tidak langsung terhubung dengan kotak komponen. Ukuaran gypsum disesuaikan dengan Hotplate.



Gambar 3.12. Peredam panas



Gambar 3.13. Keseluruhan alat

## BAB 4

### HASIL EKSPERIMEN DAN ANALISA DATA

Setelah dilakukan pengerjaan keseluruhan sistem, maka perlu dilakukan pengujian alat serta penganalisaan terhadap alat, apakah sistem sudah bekerja dengan baik atau tidak. Pengujian-pengujian tersebut meliputi :

- Pengujian Rangkaian Minimum system dan keypad
- Pengujian ADC (*Analog to Digital Conversion*)
- Pengkalibrasian Rangkaian *Cold Junction*
- Pengujian *thermocouple*
- Pengujian Rangkaian Driver Motor dan PWM
- Pengujian Sistem Kendali dengan Metode *Direct Synthesis*

#### 4.1 Pengujian Rangkaian Minimum Sistem dan Keypad

Pengujian rangkaian minimum system dan keypad ini bertujuan untuk mengetahui rangkaian tersebut dapat berfungsi atau tidak. Rangkaian minimum sistem ini sebagai bagian yang paling utama. Hasil pengujian ini dapat di lihat pada tampilan di LCD. Dalam pengendaliannya dengan menggunakan keypad, Keypad yang di gunakan keypad ukuran 4x4. Pengujiannya dengan mendownload program keypad ke minimum system dengan prosedur sebagai berikut:

1. Menyambungkan konektor catu daya 5 VDC ke rangkaian minimum system yang terhubung dengan keypad.
2. Menghubungkan Downloader ke komputer (melalui usb).
3. Menghubungkan konektor ISP pada downloader ke rangkaian keypad.
4. Mendownload program ke rangkaian mikrokontroler.
5. Menekan tombol keypad secara bergantian.

Dari hasil pengujian yang didapat dari program rangkaian minimum system berjalan dengan baik karena dapat menampilkan nilai keypad yang ditekan pada LCD. Berikut data hasil nilai saat penekanan masing-masing tombol keypad:



Tabel 4.1 Nilai keypad yang ditampilkan pada LCD

| Karakter pada Tombol keypad | Nilai keypad pada LCD |
|-----------------------------|-----------------------|
| 1                           | 0                     |
| 2                           | 1                     |
| 3                           | 2                     |
| 4                           | 4                     |
| 5                           | 5                     |
| 6                           | 6                     |
| 7                           | 8                     |
| 8                           | 9                     |
| 9                           | 10                    |
| 0                           | 13                    |
| A                           | 3                     |
| B                           | 7                     |
| C                           | 11                    |
| D                           | 15                    |
| *                           | 12                    |
| #                           | 14                    |
| Keypad tidak ditekan        | 16                    |

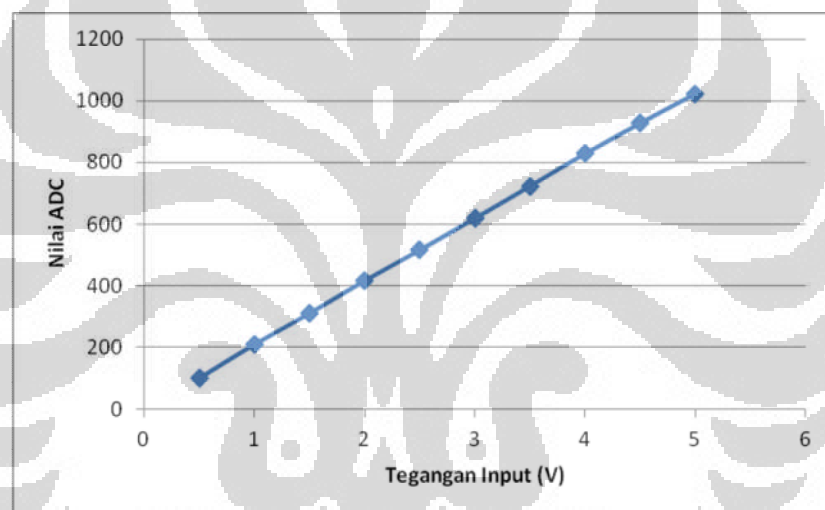
Dari hasil yang didapat pada penekanan masing-masing tombol keypad mempunyai nilai yang berbeda-beda seperti pada tabel di atas dapat disimpulkan bahwa rangkaian keypad bekerja dengan baik.

#### 4.2 Pengujian ADC (*Analog to Digital Conversion*)

Dalam pengujian ADC ini dengan cara memberi tegangan input pada ADC di ATmega 16 yang sudah diprogram untuk pembacaan ADC, kemudian mencatat data digital yang di tampilkan pada LCD. Berikut hasil pengujian ADC:

Tabel 4.2. Data pengujian ADC

| Tegangan Input | Nilai ADC |
|----------------|-----------|
| 0.5            | 100       |
| 1              | 209       |
| 1.5            | 311       |
| 2              | 416       |
| 2.5            | 517       |
| 3              | 619       |
| 3.5            | 722       |
| 4              | 829       |
| 4.5            | 927       |
| 5              | 1023      |



Gambar 4.1. Perubahan nilai ADC terhadap tegangan input

Dari grafik di atas nilai ADC yang dihasilkan linear atau berbanding lurus terhadap tegangan input yang diberikan, sehingga dapat disimpulkan ADC berjalan dengan baik.

### 4.3 Pengkalibrasian Rangkaian Cold junction

Setelah pengukuran *cold junction*, dilakukan kalibrasi rangkaian cold junction. ini bertujuan agar membuat respon antara *thermocouple* dengan LM35 sama. *Thermocouple* itu mengukur beda temperatur di ubah menjadi beda tegangan. Berarti harus mengetahui beda temperatur yang konstan terlebih dulu.

Sebelumnya rangkaian ini di offset nul. Tegangan offset keluaran (tegangan kesalahan) disebabkan oleh arus bias masukan. Bila tegangan kedua masukan sama besar, keluaran op amp akan nol volt. Namun jarang ditemukan kejadian seperti itu sehingga keluarannya akan ada sedikit tegangan. Keadaan seperti inilah dapat di atasi dengan teknik penolan offset. Namun karena rangkaiannya DC dengan menggunakan op amp cenderung lebih rentan terhadap perubahan temperatur yang dapat mengganggu keseimbangan op amp yang telah di atur sebelumnya akibatnya pada keluaran akan terjadi kesalahan. Kesalahan disini berarti mempunyai beda.

Kemudian untuk mengukur temperatur pada objek, maka sebelumnya harus diketahui suhu referensi yaitu suhu lingkungan yang diukur oleh sensor LM35 yang nantinya akan dijumlahkan dengan temperatur yang dihasilkan *Thermocouple*, maka untuk tujuan tersebut *Thermocouple* harus dikalibrasi dengan prosedur sebagai berikut:

1. Menghubungkan rangkaian cold junction ke sumber catu daya.
2. Membaca temperatur ruang dengan thermometer digital Flux.
3. Menghubungkan sensor *Thermocouple* ke terminal *Thermocouple*.
4. Memanaskan air hingga 100 derajat pada wadah yang dicelupkan sensor *Thermocouple* dan thermometer digital.
5. Mengukur beda temperatur antara temperature ruang dan temperatur yang terbaca pada thermometer digital yang tercelup pada air.
6. Membaca output cold junction dan mengatur penguat hingga output terbaca pada nilai yang ditentukan pada persamaan 4.1.

Dari prosedur diatas dapat dideskripsikan dengan awalnya mengukur suhu lingkungan atau ruang, kemudian mengukur suhu pada air yang dipanaskan misal:

Temperatur ruang ( $T_{\text{cold}}$ ) = 30°C

Temperatur air ( $T_a$ ) = 100 °C

maka output yang harus terukur saat temperatur 100 °C adalah:

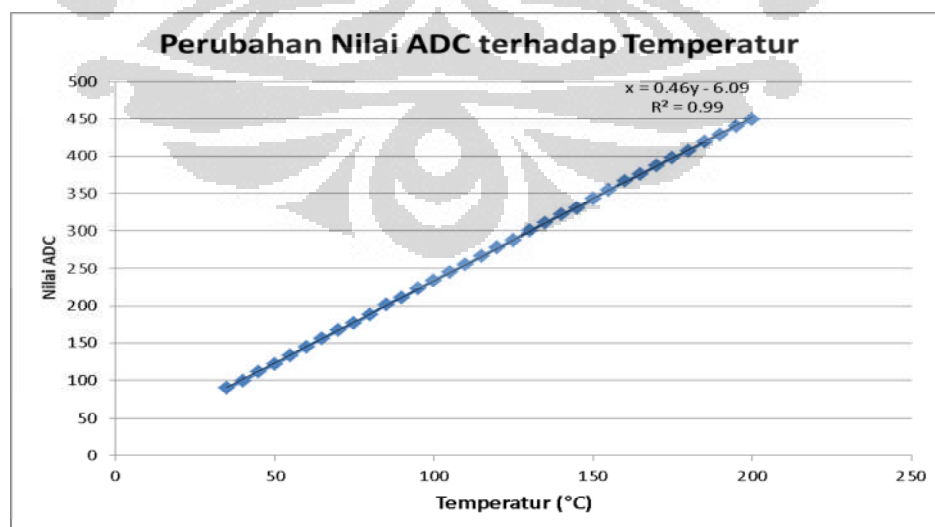
$$V_{\text{out}} = (T_a - T_{\text{cold}}) \times \frac{10\text{mV}}{^{\circ}\text{C}} \quad (4.1)$$

$$V_{\text{out}} = 700\text{mv}$$

kemudian tegangan keluaran *cold-junction* harus bernilai 700mv dengan mengubah resistansi pada Rgain (trimpot) 10KΩ. Dalam pengaturan output cold junction vout dapat diatur pada tegangan yang diinginkan sehingga dapat disimpulkan rangkaian berjalan dengan baik.

#### 4.4 Pengujian *Thermocouple*

Pengujian thermocouple dilakukan setelah rangkaian *cold junction* di kalibrasi. Pengukuran dengan menggunakan termometer digital dimana thermocouple di celupkan ke oil yang di panaskan hingga 200°C, thermocouple ini dihubungkan ke rangkaian cold junction. Rangkaian cold junction di beri masukan dengan CT 12volt. Tegangan keluaran di cold junction dihubungkan ke rangkaian minis pada ADC, yang kemudian diproses dan ditampilkan pada LCD. Program dijalankan, masukan termocouple sejajar dengan termometer digital, baca perubahan temperatur pada termometer digital dan ADC pada LCD. Pengambilan data dilakukan setiap 5 °C, Setelah melakukan percobaan didapatkan nilai bit ADC-nya sebagai berikut:



Gambar 4.2. Grafik rata-rata temperature turun

Dari percobaan yang telah dilakukan di dapat bahwa pada saat penurunan temperatur, persamaan yang di dapat dianggap linier. Hal ini terjadi karena pada saat mengkalibrasi *thermocouple* dilakukan dengan menggunakan termometer digital dimana dihubungkan dengan *thermocouple* sehingga membuat respon antara *thermocouple* dengan LM35 sama, Dari grafik terlihat bahwa perubahan temperaturnya linear dengan persamaan garis, yaitu:

$$x = 0.46y - 6.09 \quad (4.2)$$

$$R^2 = 0.99$$

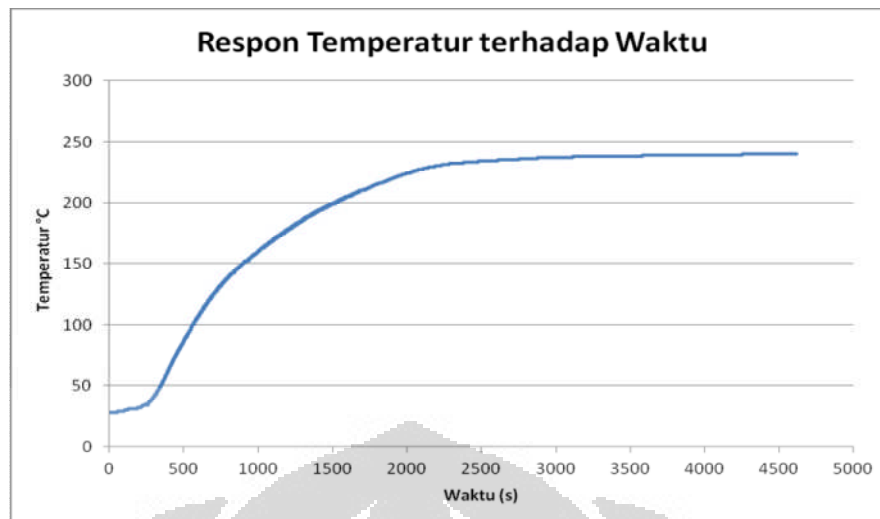
dimana :  $x = T = \text{Temperatur}$   
 $y = \text{Nilai ADC}$

Setelah didapatkan persamaan garis di atas maka dapat ditentukan berapa temperatur yang dihasilkan dengan persamaan:

$$y = \frac{T + 6.09}{0.46} \quad (4.3)$$

Dimana T adalah nilai temperatur yang terukur pada termometer digital dan x adalah nilai bit dari ADC. Dari persamaan garis didapatkan nilai  $R^2 = 0.99$ , artinya sensor temperatur yang digunakan dalam pengukuran temperaturnya dapat dikatakan baik. Data yang diperoleh ini bisa memudahkan kita dalam mendapatkan nilai persamaan yang akan dimasukkan dalam program persamaan termometer dan *thermocouple*.

Setelah didapatkan rata-rata temperatur maka diuji kembali dengan mencoba memanaskan plat heater dengan daya 70% hingga batas temperatur  $240^{\circ}\text{C}$ . Dengan demikian didapatkan grafik:



Gambar 4.3. Perubahan temperatur pada daya 70%

#### 4.5 Pengujian Rangkaian Driver Motor

Pengujian ini bertujuan untuk mengetahui apakah driver motor DC dapat berfungsi dengan baik sebagai penggerak dan untuk mengetahui kecepatan respon dari driver motor DC terhadap perubahan kecepatan. Pengujian ini meliputi pengujian driver motor dan pengujian kecepatan motor.

##### 4.3.1 Pengujian Driver Motor

Dalam pengujian ini hal yang perlu dilakukan adalah, pertama beri tegangan 12Volt pada konektor rangkaian penggerak yang terhubung dengan relay, pasang motor pada rangkaian driver dan beri tegangan 5Volt pada kaki anoda optokopler pada kaki yang akan dikoneksikan ke mikrokontroler sebagai PWM. Kemudian hubungkan ground ke kaki katoda optocoupler. Bila logika '0' diberikan pada kaki PWM maka motor akan bergerak .

##### 4.3.2 Pengujian Kecepatan Motor

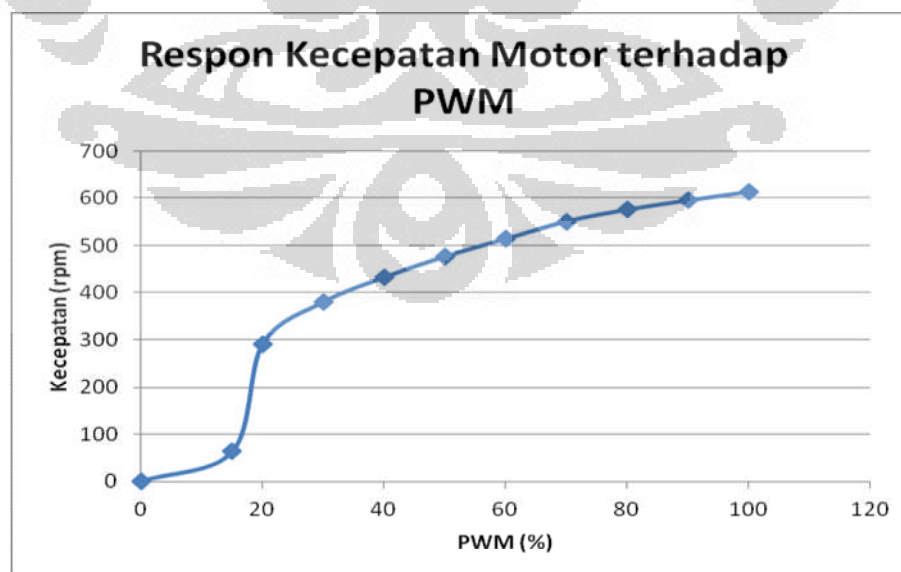
Selanjutnya menguji kecepatan motor dengan prinsip kerja PWM. Dengan mendownload program PWM dan memasukan *set point* melalui keypad (nilai set point dari 0% - 100%). Pengukuran kecepatan motor menggunakan alat ukur tachometer digital tipe dekko dt 2234L, dengan prosedur pengujian sebagai berikut:

1. Memberikan kertas putih pada magnet yang berputar.
2. Memasukan nilai set point (nilai PWM), kemudian jalankan motor.
3. Mengarahkan tachometer pada bagian magnet yang berputar pada bagian kertas putih dan tahan tachometer selama 30 detik, kemudian baca data tachometer.

Dari pengujian, motor DC beroperasi dengan baik dengan hasil pengujian PWM sebagai berikut:

Tabel 4.3. Data hasil pengujian kecepatan motor terhadap PWM

| PWM (%) | Kecepatan (rpm) |        |        | Kecepatan rata-rata (rpm) |
|---------|-----------------|--------|--------|---------------------------|
|         | data 1          | data 2 | data 3 |                           |
| 0       | 0               | 0      | 0      | 0                         |
| 15      | 60              | 66     | 63     | 63                        |
| 20      | 285             | 297    | 291    | 291                       |
| 30      | 376             | 384    | 380    | 380                       |
| 40      | 432             | 435    | 431    | 433                       |
| 50      | 475             | 476    | 480    | 477                       |
| 60      | 513             | 515    | 512    | 513                       |
| 70      | 546             | 559    | 545    | 550                       |
| 80      | 573             | 577    | 575    | 575                       |
| 90      | 595             | 596    | 593    | 595                       |
| 100     | 610             | 613    | 615    | 613                       |



Gambar 4.4. Respon kecepatan motor terhadap PWM

Dari grafik diatas terlihat bahwa kecepatan putaran motor mulai linear saat pemberian PWM diatas 20%, ini dikarenakan adanya penggunaan gear box sehingga daerah kerja motor bekerja dengan baik pada pemberian PWM diatas 20%.

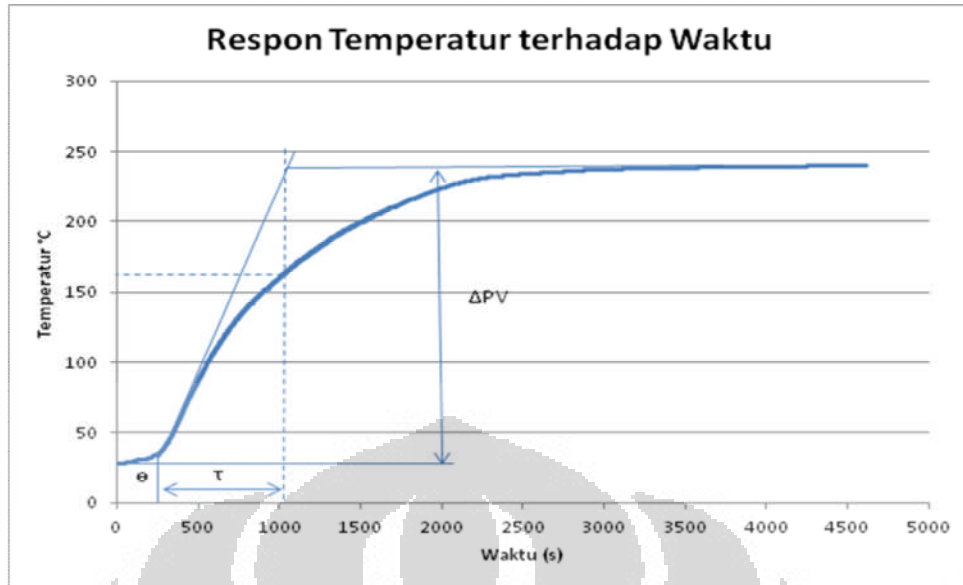
#### 4.5 Pengujian Sistem Pengendalian Dengan Metode *Direct Synthesis*

Pengujian sistem pengendali dengan metode *direct synthesis* berfungsi mencari fungsi tranfers dengan mencoba sistem tersebut. Dari design Grafik daya 70% pada pengujian *thermocouple* merupakan langkah awal yang dilakukan untuk menentukan nilai matematik dari respon sistem, untuk mendapatkan fungsi tranfersnya harus mencari nilai  $\theta$  dan nilai  $\tau$ . Untuk mendapatkan nilai  $\theta$ , yakni dengan menarik garis lurus pada saat grafik mengalami kenaikan yang secara konstan. Kemudian ditarik garis lurus horisontal untuk mendapatkan titik temu dengan garis lurus pada grsfik. Pada titik temu tersebut ditarik garis vertikal hingga diketahui posisi garis vertikal tersebut. Nilai  $\theta$  adalah nilai yang ditunjukkan garis tersebut. Sedangkan untuk Mendapatkan nilai  $\tau$  harus mendapatkan sebuah titik terlebih dahulu yakni dengan perhitungan seperti dibawah ini :

$$\text{Titik} = PV1 + (0,632 \times \Delta PV) \quad (4.4)$$

Dimana PV1 adalah batas minimum temperatur pada Gambar 4.6 dan  $\Delta PV$  adalah nilai selisih antara batas maksimum temperatur dengan batas minimum temperatur pada Grafik Dengan melakukan perhitungan seperti persamaan (4.4) maka didapatkan nilai titik tersebut yakni 1014. Kemudian pada titik tersebut ditarik garis lurus horisontal hingga menyentuh grafik dan menghasilkan titik temu. Pada titik pertemuan tersebut ditarik garis lurus vertikal hingga diketahui posisi garis vertikal tersebut. Nilai  $\tau$  adalah selisih dari nilai yang ditunjukkan garis tersebut dengan nilai  $\theta$ . Untuk mencari nilai  $\theta$  dan nilai  $\tau$  juga dapat melihat Gambar 4.6.





Gambar 4.5. Mencari Nilai  $\theta$  dan Nilai  $\tau$

Bila dilihat dari gambar diatas, dapat diketahui fungsi transfer dari system lalu dengan menggunakan metode *direct synthesis* dapat diperoleh nilai  $K_p$ ,  $T_i$  dan  $T_d$ . Sebelum itu, dari gambar diatas diperoleh nilai  $\theta = 212$  dan nilai  $\tau = 1014 - 212 = 802$ . Sedangkan nilai  $\Delta MV = 70\% - 10\% = 60\%$  dan  $\Delta PV = 240 - 28 = 212$  dengan batas maksimum temperatur  $\Delta PV = (212 / 300) \times 100\% = 70,6\%$ .

Dengan demikian persamaan yang didapat :

$$MV = K_p \left( E + \frac{1}{T_i} \int E dt + T_d \frac{dE}{dt} \right) \quad (4.5)$$

Dari persamaan diatas dapat mencari nilai  $K_p$ ,  $T_i$ , dan  $T_d$  dengan menggunakan persamaan-persamaan dibawah ini :

$$T_i = \frac{\theta}{2} + \tau$$

$$T_i = 908s$$

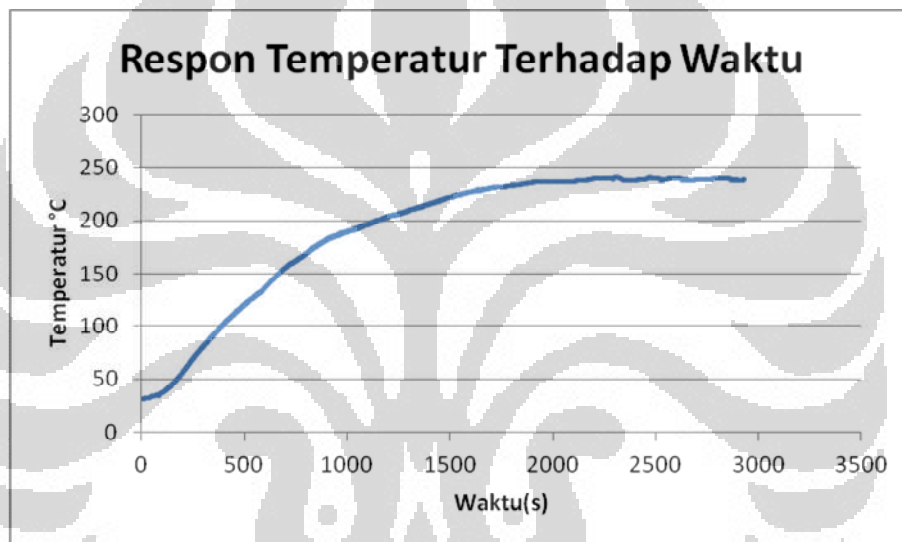
$$T_d = \frac{\tau}{2\left(\frac{\tau}{\theta}\right) + 1}$$

$$T_d = 93,6s$$

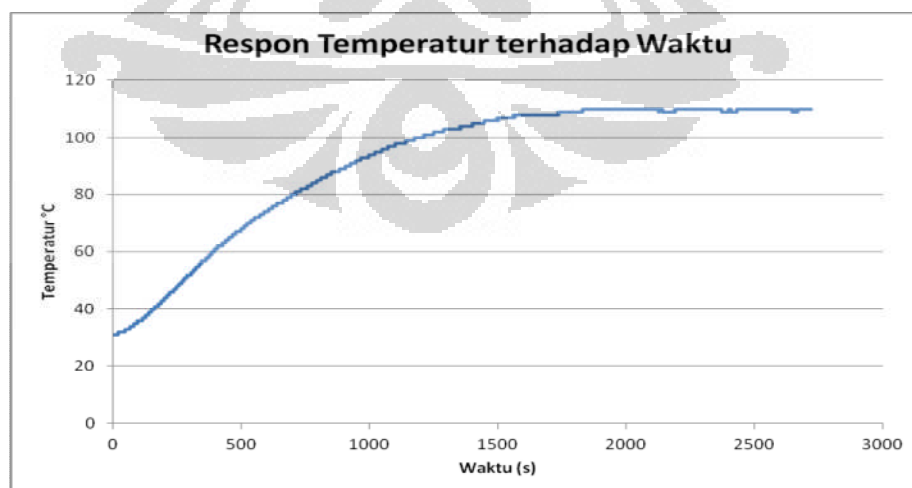
$$K_p = \frac{2\left(\frac{\tau}{\theta}\right) + 1}{2K\left(\frac{\tau_c}{\theta}\right) + 1}$$

$$K_p = 0.81$$

Setelah itu, nilai dari persamaan-persamaan diatas yang telah didapatkan kemudian dimasukkan kedalam program pengendalian. Kemudian diuji dengan set point 240 °C dan 110 °C ,didapatkan data seperti yang tertera pada Gambar 4.7



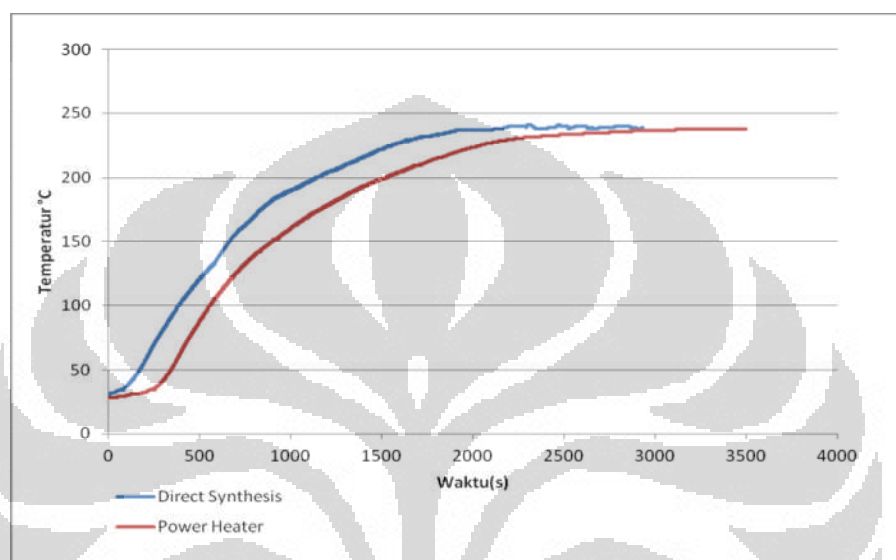
Gambar 4.6. Respon temperatur terhadap waktu pada *set point* 240°C



Gambar 4.7. Respon temperatur terhadap waktu pada *set point* 110°C

Berdasarkan dari data tersebut untuk percobaan set point (SP) = 240°C dan 110C berjalan cukup baik dan dapat diketahui bahwa nilai temperatur yang mengalami sedikit *error steadystate* hingga 2°C.

Dapat diambil analisa, contoh untuk pengendalian dengan power heater 70% dan dengan pengendalian metode direct synthesis dapat dilihat pada Gambar 4.9.



Gambar 4.8. Perbandingan respon system dengan pengendalian PID dan tanpa pengendalian PID.

Dari Gambar 4.9. nampak bahwa waktu untuk pencapaian set point lebih cepat dengan pengendalian PID. Sehingga pengendalian dengan PID metode direct synthesis lebih baik dari pada tanpa pengendalian PID.

## BAB 5

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Setelah menyelesaikan penelitian Rancang Bangun Hotplate Stirrer Magnetik Terkendali Temperatur dan Kecepatan Pengaduk serta pengambilan data, maka penulis dapat menarik kesimpulan bahwa :

- Alat ini berjalan cukup baik dalam proses pemanasan dan pengadukan dengan mengatur temperatur dan kecepatan pengaduk.
- Persamaan dari nilai ADC terhadap perubahan temperatur yang di peroleh adalah  $T = 0.46x - 6.09$ , yang digunakan untuk mengkonversi nilai ADC menjadi nilai Temperatur.
- Persamaan konversi dari data ADC diperoleh nilai  $R^2 = 0.99$  yang berarti bahwa sensor suhu bekerja dengan baik.
- Pada pengendalian kecepatan pengaduk, respon pemberian *set point* sesuai yang diinginkan yaitu semakin besar pemberian *setpoint*, semakin cepat perputaran kecepatan pengaduknya.
- Daerah kerja kecepatan motor akan linear saat *set point* PWM diatas 20%.
- Respon pengendalian PID yang dihasilkan dengan metoda *direct synthesis method* berjalan baik, dengan *error steady state* sebesar 2 °C.
- Hasil dengan menggunakan *direct synthesis method* didapat persamaan pengendali berupa pengendali PID dengan nilai  $K_p = 0,81$  ;  $T_i = 908$  dan  $T_d = 93,6$ .
- Hasil dengan menggunakan pengendalian PID *direct synthesis method* pada saat pencapaian nilai set point lebih cepat dari pada tanpa penggunaan pengendalian PID (dengan mengatur power heater).

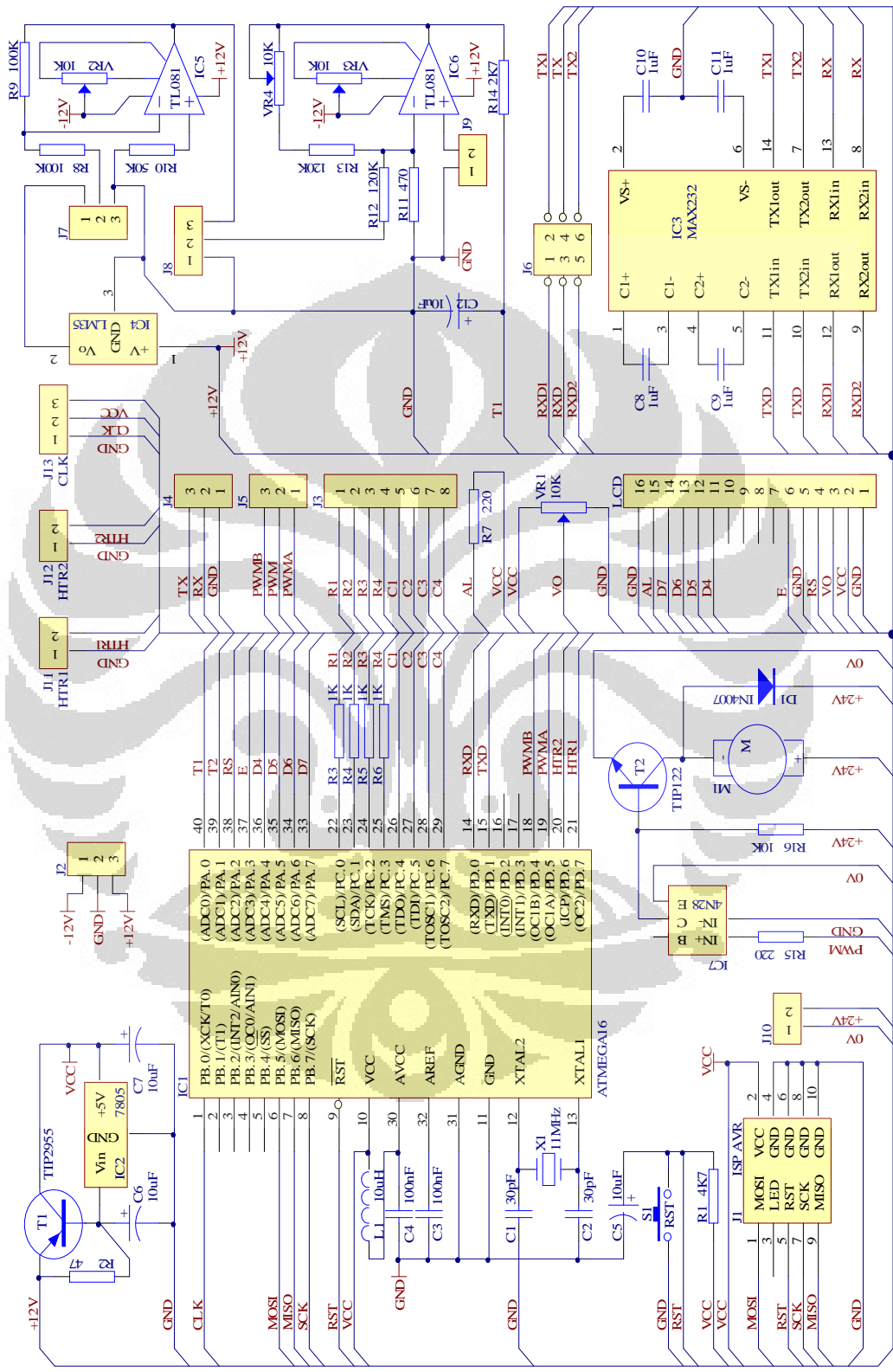
#### 5.2 Saran

- Sebaiknya tidak menyentuh resistor pengatur tegangan offset yang sudah dikalibrasi, karena rentan terhadap pengukuran temperatur.

## DAFTAR PUSTAKA

- [1] Kilian, Christopher T. (2001). *Modern Control Technology : Components and Systems*. Delmar.
- [2] Hapsah, (2009), *Rancang Bangun Automatic Mixer Temperatur dan Kecepatan Pengaduk Terkendali*, Dalam Laporan Tugas akhir, Depok: Departemen Fisika, Universitas Indonesia.
- [3] Sumanto.(1996), *Mesin Arus Searah*, Yogyakarta.
- [4] Eko,Putra, & Agfianto Eko. (2003), *Belajar Mikrokontroller*, Jakarta: Gava Media.
- [5] Hariadi, Eko. (2004), *Kemagnetan dan Induksi Elektromagnetik*, Jakarta
- [6] Ogata, Katsuhiko.(1985). *Teknik Kontrol Automatik*. Erlangga.
- [7] Wicaksono, Handy. (2004), *Analisa Performansi dan Robustness Beberapa Metode Tuning Kontroler PID pada Motor DC*, September 2, 2004, <http://puslit2.petra.ac.id/ejournal/index.php/elk/article/viewFile/16191/16183>

# Lampiran 1 : Rangkaian Sistem Alat



**Lampiran 3 : Tabel data kalibrasi Termocouple**

| Temperature | Nilai ADC |     |     |     |     | Rata-rata |
|-------------|-----------|-----|-----|-----|-----|-----------|
|             | 1         | 2   | 3   | 4   | 5   |           |
| 200         | 445       | 442 | 442 | 444 | 443 | 443.2     |
| 195         | 433       | 434 | 433 | 435 | 434 | 433.8     |
| 190         | 425       | 426 | 425 | 426 | 426 | 425.6     |
| 185         | 416       | 417 | 417 | 417 | 416 | 416.6     |
| 180         | 407       | 409 | 407 | 408 | 407 | 407.6     |
| 175         | 398       | 399 | 398 | 398 | 399 | 398.4     |
| 170         | 390       | 392 | 391 | 390 | 390 | 390.6     |
| 165         | 379       | 381 | 381 | 380 | 380 | 380.2     |
| 160         | 369       | 370 | 369 | 369 | 370 | 369.4     |
| 155         | 359       | 360 | 358 | 359 | 360 | 359.2     |
| 150         | 348       | 350 | 348 | 348 | 348 | 348.4     |
| 145         | 336       | 338 | 336 | 336 | 337 | 336.6     |
| 140         | 324       | 324 | 325 | 325 | 323 | 324.2     |
| 135         | 313       | 314 | 313 | 313 | 313 | 313.2     |
| 130         | 303       | 305 | 303 | 303 | 304 | 303.6     |
| 125         | 290       | 290 | 291 | 290 | 290 | 290.2     |
| 120         | 278       | 278 | 279 | 278 | 279 | 278.4     |
| 115         | 268       | 269 | 268 | 269 | 269 | 268.6     |
| 110         | 259       | 260 | 258 | 259 | 259 | 259       |
| 105         | 248       | 249 | 249 | 249 | 248 | 248.6     |
| 100         | 239       | 239 | 240 | 239 | 239 | 239.2     |
| 95          | 227       | 228 | 228 | 228 | 228 | 227.8     |
| 90          | 215       | 216 | 215 | 215 | 216 | 215.4     |
| 85          | 204       | 205 | 204 | 205 | 205 | 204.6     |
| 80          | 185       | 187 | 186 | 186 | 185 | 185.8     |
| 75          | 175       | 177 | 177 | 177 | 176 | 176.4     |
| 70          | 164       | 165 | 165 | 165 | 165 | 164.8     |
| 65          | 154       | 154 | 155 | 154 | 154 | 154.2     |
| 60          | 145       | 145 | 145 | 146 | 145 | 145.2     |
| 55          | 133       | 134 | 134 | 134 | 135 | 134       |
| 50          | 120       | 122 | 120 | 120 | 122 | 120.8     |
| 45          | 110       | 110 | 111 | 109 | 110 | 110       |
| 40          | 99        | 100 | 99  | 99  | 98  | 99        |
| 35          | 88        | 88  | 88  | 89  | 88  | 88.2      |

## Lampiran 4 : Program

```
'-----PROGRAM HOTPLATE STIRRER-----'  
'----- Ekstensi Instrument UI -----'  
'-----'
```

```
$regfile = "m16def.dat"  
$crystal = 11059200  
$baud = 115200
```

```
On Ovfl Time_base  
On Urxc Data_seri_in
```

```
Config Timer0 = Counter , Edge = Falling  
Config Timer1 = Timer , Prescale = 8  
Config Lcdpin = Pin , Db4 = Porta.4 , Db5 = Porta.5 , Db6 = Porta.6 ,  
Db7 = Porta.7 , E = Porta.3 , Rs = Porta.2  
Config Lcd = 20 * 4  
Config Adc = Single , Prescaler = Auto , Reference = Avcc  
Config Kbd = Portc.debounce = 40  
Config Portd.4 = Output  
Config Portd.5 = Output  
Config Portd.6 = Output  
Config Portd.7 = Output
```

```
Motor1 Alias Portd.4  
Motor2 Alias Portd.5  
Heater1 Alias Portd.7  
Heater2 Alias Portd.6
```

```
Dim Ulang As Bit  
Dim Stop_flag As Bit  
Dim Setting_temp As Bit  
Dim Setting_mtr As Bit  
Dim Setting_time As Bit  
Dim Mode_move As Bit  
Dim Periode_1s_flag As Bit  
Dim Periode_2s_flag As Bit  
Dim Kirim_data_flag As Bit  
Dim Start_send_flag As Bit  
Dim Stop_data_flag As Bit  
Dim Stop_program As Bit  
Dim Balik As Bit  
Dim Periode_12_flag As Bit  
Dim Keypad As Byte  
Dim Periode_1s As Byte  
Dim Periode_10ms As Byte  
Dim Periode_hz As Byte  
Dim Nilai_keypad As Byte  
Dim Power_heater As Byte  
Dim Power_motor As Byte  
Dim Data_set2 As Byte  
Dim Iterasi As Byte  
Dim Mv_htr As Byte  
Dim Mv_mtr As Byte
```



```

Dim Temperature As Byte
Dim Periode_1d As Word
Dim Value_set As Word
Dim Periode As Word
Dim Nilai_adc As Word
Dim Waktu As Word
Dim Rpm_lcd As Word
Dim Periode_2d As Word
Dim Rotasi As Word
Dim Rotasi1 As Word
Dim Stirrer As Single
Dim Rpm As Single
Dim Nilai_temp As Single
Dim Nilai_temp_old As Single
Dim Error As Single
Dim Error_old As Single
Dim Kp As Single
Dim Ti As Single
Dim Td As Single
Dim Sp As Single
Dim Pv As Single
Dim Mv As Single
Dim Mv_p As Single
Dim Mv_i As Single
Dim Mv_d As Single
Dim Sigma_error_kp As Single
Dim Wind_up_limit As Single
Dim Delta_error As Single
Dim C As Single
Dim Temp_lcd As Integer
Dim Data_seri As String * 1

Enable Interrupts
Enable Ovfl
Enable Urxc

$eeprom
Data_controller:
Data 30 , 30 , 30 , 30 , 30
$data

Reset Motor1
Reset Heater1

Start_send_flag = 0
Counter1 = 65280
Counter0 = 128

Ulang = 1
Setting_temp = 0
Setting_mtr = 0
Periode_10ms = 0
Periode_1s = 1
Stop_data_flag = 0
Periode_1s_flag = 0

Deflcdchar 0 , 16 , 16 , 16 , 16 , 16 , 16 , 16 , 16
Main_program:

```

```

Stop_program = 0
Stop_flag = 1
Mv_htr = 0
Mv_mtr = 0
Temperature = 0
Periode = 10
Gosub Write_data
Cls
Wait 1
Locate 1 , 1
Lcd " HOTPLATE STIRRER"
Locate 2 , 1
Lcd "  TEKAN!!!  "
Locate 3 , 1
Lcd " (1) Mode Auto  "
Locate 4 , 1
Lcd " (2) Mode Manual "
Cursor Off Noblink
Ulang = 1
Do
Keypad = Getkbd()
If Keypad < 16 Then
  Select Case Keypad
  Case 0
    Nilai_keypad = 1
  Case 1
    Nilai_keypad = 2
  End Select
  Waitms 100
  Balik = 1
  Do
    Keypad = Getkbd()
    If Keypad = 16 Then Balik = 0
  Loop Until Balik = 0
  End If
  Waitms 100
  If Nilai_keypad = 1 Then
    Mode_move = 0      'auto
    Gosub Set_temp
    Gosub Set_motor
    Gosub Set_wktu
    Ulang = 0
  Else
    If Nilai_keypad = 2 Then      'manual
      Mode_move = 1
      Gosub Set_temp
      Gosub Set_motor
      Gosub Set_wktu
      Ulang = 0
    End If
  End If
Loop Until Ulang = 0
Ulang = 1
*****

Start_program:
Waktu = 60 * Periode
Stop_flag = 1

```

```

Start Adc

If Mode_move = 1 Then
  Gosub Tampilan_manual
  Waitms 100

Do
  If Periode_1s_flag = 1 Then
    Periode_1s_flag = 0
    If Waktu = 0 Then
      Stop_flag = 0
    Else
      Waktu = Waktu - 1
    End If
    Gosub Baca_adc
    Gosub Baca_motor

  End If
  -----
  If Start_send_flag = 1 Then
    If Stop_data_flag = 1 Then
      Start_send_flag = 0
      Stop_data_flag = 0
    End If
    If Kirim_data_flag = 1 Then
      Kirim_data_flag = 0
      Print Temp_lcd ; ":" ; Rpm ; ":" ; Waktu ; "#"
    End If
  End If
  -----
  Gosub Keypad1
  Waitms 100
Loop Until Stop_flag = 0

Else
  Gosub Tampilan_auto
  Error = 0
  Sigma_error_kp = 0
  Temp_lcd = 0
  Kp = 0.81
  Ti = 908
  Td = 93.6

Do
  If Periode_2s_flag = 1 Then
    Periode_2s_flag = 0
    Error_old = Error
    Gosub Baca_adc
    ' C = Temperature / 100
    Sp = Temperature / 2.5
    Pv = Nilai_temp / 2.5
    Error = Sp - Pv
    Mv_p = Kp * Error
    Sigma_error_kp = Sigma_error_kp + Mv_p
    Wind_up_limit = 100 * Ti
    Delta_error = Error - Error_old
    If Sigma_error_kp < 0 Then Sigma_error_kp = 0
    If Sigma_error_kp > Wind_up_limit Then Sigma_error_kp = Wind_up_limit
  End If
End Do

```

```

Mv_i = Sigma_error_kp / Ti
Mv_d = Kp * Delta_error
Mv_d = Mv_d * Td
Mv = Mv_p + Mv_i
Mv = Mv + Mv_d
If Mv > 100 Then Mv = 100
If Mv < 0 Then Mv = 0
Mv_htr = Mv
End If
If Periode_1s_flag = 1 Then
  Periode_1s_flag = 0
  If Waktu = 0 Then
    Stop_flag = 0
  Else
    Waktu = Waktu - 1
  End If
  Gosub Baca_motor
End If
If Periode_12_flag = 1 Then
  Periode_12_flag = 0
  If Rpm_lcd > Rotasi Then
    Mv_mtr = Mv_mtr
    Rotasi1 = Rotasi + 20
    If Rpm_lcd > Rotasi1 Then
      Mv_mtr = Mv_mtr - 1
    End If
  Else
    Mv_mtr = Mv_mtr + 1
  End If
End If
-----
If Start_send_flag = 1 Then
  If Stop_data_flag = 1 Then
    Start_send_flag = 0
    Stop_data_flag = 0
  End If
  If Kirim_data_flag = 1 Then
    Kirim_data_flag = 0
    Print Temp_lcd ; ":" ; Mv_mtr ; ":" ; Waktu ; "#"
  End If
End If
-----
  Gosub Keypad1
Loop Until Stop_flag = 0

End If
Goto Main_program

*****

Set_temp:
  Iterasi = 0
  Setting_temp = 0
  Ulang = 1
  Cls
  Locate 1 , 1
  Lcd "SETTING TEMPERATURE"

```

```

If Mode_move = 0 Then
  Locate 2 , 1
  Lcd "  Mode auto  "
  Locate 3 , 9
  Lcd Mv_htr
  Locate 4 , 1
  Lcd "Nilai Maximal 250 " ; Chr(223) ; "C"
  Locate 3 , 9
  Cursor On Blink
  Waitms 200
Else
  Locate 2 , 1
  Lcd "  Mode manual  "
  Locate 3 , 9
  Lcd Mv_htr
  Locate 4 , 1
  Lcd "Nilai Maximal 100 % "
  Locate 3 , 9
  Cursor On Blink
  Waitms 200
End If
Do
  Keypad = Getkbd()
If Keypad < 16 Then
  Gosub Keypad2
  Balik = 1
  Do
    Keypad = Getkbd()
    If Keypad = 16 Then Balik = 0
  Loop Until Balik = 0
  If Nilai_keypad < 10 Then
    Iterasi = Iterasi + 1
    If Iterasi = 1 Then
      Value_set = Nilai_keypad
    Else
      Value_set = 10 * Value_set
      Value_set = Value_set + Nilai_keypad
    End If
  End If
  If Mode_move = 0 Then
    'auto
    If Value_set > 250 Then
      Temperature = 250
      Mv_htr = 0
    Else
      Mv_htr = 0
      Temperature = Value_set
    End If
  Else
    'manual
    If Value_set > 100 Then
      Mv_htr = 100
      Temperature = 0
    Else
      Mv_htr = Value_set
      Temperature = 0
    End If
  End If
  If Nilai_keypad = 10 Then
    Gosub Write_data

```

```

    Ulang = 0
    Cursor Off Noblink
    If Mode_move = 0 Then
        Gosub Tampilan_auto
    Else
        Gosub Tampilan_manual
    End If
End If
End If
Loop Until Ulang = 0
Ulang = 1
Gosub Read_data
Return

```

```

Set_motor:
    Iterasi = 0
    Setting_mtr = 0
    Ulang = 1
    Cls
    Locate 1 , 1
    Lcd " SETTING STIRRER "

    Locate 2 , 1
    Lcd " Kecepatan Motor "
    Locate 3 , 9
    Lcd " ___ "
    Locate 4 , 1
    Lcd " Nilai Max PWM 100% "
    Locate 3 , 9
    Cursor On Blink
    Waitms 200

Do
    Keypad = Getkbd()
    If Keypad < 16 Then
        Gosub Keypad2
        Balik = 1
    Do
        Keypad = Getkbd()
        If Keypad = 16 Then Balik = 0
    Loop Until Balik = 0
    If Nilai_keypad < 10 Then
        Iterasi = Iterasi + 1
        If Iterasi = 1 Then
            Value_set = Nilai_keypad
        Else
            Value_set = 10 * Value_set
            Value_set = Value_set + Nilai_keypad
        End If
    End If
End If

If Mode_move = 0 Then
    If Value_set > 900 Then
        Rotasi = 900
        Mv_mtr = 0
    Else
        Mv_mtr = 0
    End If

```

```

    Rotasi = Value_set
  End If
Else
  If Value_set > 100 Then          'manual
    Mv_mtr = 100
  Else
    Mv_mtr = Value_set
  End If
End If

If Nilai_keypad = 10 Then
  Gosub Write_data
  Ulang = 0
  Cursor Off Noblink
  If Mode_move = 0 Then
    Gosub Tampilan_auto
  Else
    Gosub Tampilan_manual
  End If
End If
End If
Loop Until Ulang = 0
Ulang = 1
Gosub Read_data
Return
+++++
+
Set_wktu:
  Iterasi = 0
  Setting_time = 0
  Ulang = 1
  Cls
  Locate 1 , 1
  Lcd "  SETTING TIMER  "
  Locate 3 , 9
  Lcd "----"
  Locate 4 , 1
  Lcd " Nilai Max 1000mnt "
  Locate 3 , 9
  Cursor On Blink

Do
  Keypad = Getkbd()
If Keypad < 16 Then
  Gosub Keypad2
  Balik = 1
  Do
    Keypad = Getkbd()
    If Keypad = 16 Then Balik = 0
  Loop Until Balik = 0
  If Nilai_keypad < 10 Then
    Iterasi = Iterasi + 1
    If Iterasi = 1 Then
      Value_set = Nilai_keypad
    Else
      Value_set = 10 * Value_set
      Value_set = Value_set + Nilai_keypad

```

```

End If
End If

If Value_set > 1000 Then
  Periode = 1000
Else
  Periode = Value_set
End If

If Nilai_keypad = 10 Then
  Gosub Write_data
  Ulang = 0
  Cursor Off Noblink
  If Mode_move = 0 Then
    Gosub Tampilan_auto
  Else
    Gosub Tampilan_manual
  End If
End If
End If
Loop Until Ulang = 0
Ulang = 1
Gosub Read_data
Return
'+++++
'*****
Data_seri_in:
Disable Interrupts
Data_seri = Inkey()
If Data_seri = "*" Then
  Data_seri = Waitkey()
  If Data_seri = "R" Then Start_send_flag = 1
  If Data_seri = "G" Then Kirim_data_flag = 1
  If Data_seri = "S" Then Stop_data_flag = 1
End If
Enable Interrupts
Return

Time_base:
Disable Interrupts
Counter1 = 65280
Periode_10ms = Periode_10ms + 1
If Periode_10ms = 54 Then
  Periode_10ms = 0
  Periode_1s = Periode_1s + 1
  If Power_heater = 0 Then
    Reset Heater1
    Reset Heater2
  Else
    Power_heater = Power_heater - 1
    Set Heater1
    Set Heater2
  End If
If Periode_1s = 100 Then
  Periode_1s_flag = 1
  Periode_1s = 0

```



```

    Power_heater = Mv_htr
  End If
End If

Periode_hz = Periode_hz + 1
If Power_motor = 0 Then
  Set Motor1
  Set Motor2
Else
  Power_motor = Power_motor - 1
  Reset Motor1
  Reset Motor2
End If
If Periode_hz = 100 Then
  Periode_hz = 0
  Power_motor = Mv_mtr
End If
Periode_1d = Periode_1d + 1
If Periode_1d = 54 Then
  Periode_1d = 0
  Stirrer = Counter0
  Counter0 = 0
End If
Periode_2d = Periode_2d + 1
If Periode_2d = 2700 Then
  Periode_2d = 0
  Periode_2s_flag = 1
  Periode_12_flag = 1
End If
Enable Interrupts
Return

'*****'
Write_data:
  Writeeprom Temperature , Data_controller
  Writeeprom Mv_htr
  Writeeprom Mv_mtr
  Writeeprom Rotasi
  Writeeprom Periode
Return

Read_data:
  Readeeprom Temperature , Data_controller
  Readeeprom Mv_htr
  Readeeprom Mv_mtr
  Readeeprom Rotasi
  Readeeprom Periode
Return

Baca_adc:
  Nilai_adc = Getadc(0)
  Nilai_temp = Nilai_adc * 0.4558
  Nilai_temp = Round(nilai_temp)
  Temp_lcd = Nilai_temp
  Locate 2 , 14
  Lcd " "
  Locate 2 , 14

```

```
Lcd "" ; Temp_lcd  
Return
```

Baca\_motor:

```
Rpm = Stirrer * 100  
Rpm = Rpm / 512  
Rpm = Rpm * 60  
Rpm = Round(rpm)  
Rpm_lcd = Rpm  
Locate 3 , 14  
Lcd " "  
Locate 3 , 14  
Lcd "" ; Rpm_lcd  
Locate 4 , 14  
Lcd " "  
Locate 4 , 14  
Lcd "" ; Waktu  
Return
```

Tampilan\_manual:

```
Cls  
Locate 1 , 1  
Lcd "HOTPLATE STIRRER"  
Locate 1 , 18  
Lcd "(M)"  
Locate 2 , 1  
Lcd "PWR HTR:"  
Locate 2 , 12  
Lcd "% " ; Chr(0)  
Locate 2 , 18  
Lcd Chr(223) ; "C"  
Locate 3 , 1  
Lcd "PWR MTR:"  
Locate 3 , 12  
Lcd "% " ; Chr(0)  
Locate 3 , 18  
Lcd "rpm"  
Locate 4 , 1  
Lcd "Time :"  
Locate 4 , 13  
Lcd Chr(0)  
Locate 4 , 18  
Lcd "mnt"  
Cursor Off Noblink  
Locate 2 , 9  
Lcd Mv_htr  
Locate 3 , 9  
Lcd Mv_mtr  
Locate 4 , 9  
Lcd Periode  
Return
```

Tampilan\_auto:

```
Cls  
Locate 1 , 1  
Lcd "HOTPLATE STIRRER"  
Locate 1 , 18  
Lcd "(A)"
```

```

Locate 2 , 1
Lcd "HEATER : "
Locate 2 , 13
Lcd Chr(0)
Locate 2 , 18
Lcd Chr(223) ; "C"
  Locate 3 , 1
  Lcd "PWR MTR:"
  Locate 3 , 12
  Lcd "%" ; Chr(0)
  Locate 3 , 18
  Lcd "rpm"
  Locate 4 , 1
  Lcd "Time  :"
  Locate 4 , 13
  Lcd Chr(0)
  Locate 4 , 18
  Lcd "mnt"
  Cursor Off Noblink
  Locate 2 , 9
  Lcd Temperature
  Locate 3 , 9
  Lcd Mv_mtr
  Locate 4 , 9
  Lcd Periode
Return

```

Keypad2:

```

  Select Case Keypad
  Case 13
    Nilai_keypad = 0
  Case 0
    Nilai_keypad = 1
  Case 1
    Nilai_keypad = 2
  Case 2
    Nilai_keypad = 3
  Case 4
    Nilai_keypad = 4
  Case 5
    Nilai_keypad = 5
  Case 6
    Nilai_keypad = 6
  Case 8
    Nilai_keypad = 7
  Case 9
    Nilai_keypad = 8
  Case 10
    Nilai_keypad = 9
  Case 14
    Nilai_keypad = 10
  End Select
  If Nilai_keypad <= 9 Then
  If Iterasi = 0 Then
    Locate 3 , 9
    Lcd Nilai_keypad
  Else

```

```

    If Iterasi = 1 Then
        Locate 3 , 10
        Lcd Nilai_keypad
    Else
        Locate 3 , 11
        Lcd Nilai_keypad
    End If
End If
End If
Waitms 100
Return

Keypad1:
Keypad = Getkbd()
If Keypad < 16 Then
    Select Case Keypad
        Case 3
            Stop_program = 1
        Case 7
            Setting_temp = 1
        Case 11
            Setting_mtr = 1
        Case 15
            Setting_time = 1
    End Select
    Ulang = 0
End If
If Stop_program = 1 Then Stop_flag = 0
If Setting_temp = 1 Then Gosub Set_temp
If Setting_mtr = 1 Then Gosub Set_motor
Return

```