

**SISTEM PENGENDALI TEMPERATUR DAN KECEPATAN  
*SHAKING WATER BATH* BERBASIS *MICROCONTROLLER***

diajukan sebagai persyaratan untuk  
menyelesaikan Pendidikan Diploma 3 program Instrumentasi Elektronika

Oleh :

**Rezky Refyandini**

**2305210212**



**PROGRAM DIPLOMA 3 FISIKA INSTRUMENTASI  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS INDONESIA  
DEPOK  
2008**

**LEMBAR PENGESAHAN**

Nama : Rezky Refyandini  
NPM : 2305210212  
Jurusan : D3 Fisika Instrumentasi Elektronika  
Tanggal Sidang : 10 Juli 2008  
Judul :

**Sistem Pengendali Temperatur dan Kecepatan *Shaking*  
*Water Bath* berbasis *Microcontroller***

Laporan Tugas Akhir ini telah diperiksa, disetujui dan disahkan oleh :

**PEMBIMBING**



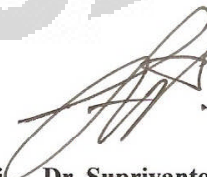
**Arief Sudarmaji, M.T**

**PENGUJI I**



**Drs. Lingga Hermanto, M.si**

**PENGUJI II**



**Dr. Supriyanto**

**PENGUJI III**



**Drs. Djati Handoko, M.Si**

*“Bismillahirrahmaanirrahiim“*

*“Sesungguhnya dalam penciptaan langit dan bumi, dan silih bergantinya malam dan siang terdapat tanda-tanda bagi orang yang berakal, (yaitu) Orang-orang yang mengingat ALLAH sambil berdiri atau duduk atau dalam keadaan berbaring dan mereka memikirkan tentang penciptaan langit dan bumi (seraya berkata), “Ya Tuhan kami, tiadalah Engkau menciptakan ini dengan sia-sia. Maha Suci Engkau, maka peliharalah kami dari siksa neraka.”*

*(Qs. Al Imran :190-191)*

## KATA PENGANTAR

Puji syukur penulis ucapkan atas Kehadirat Allah SWT, karena Rahmat dan Ridho-Nya penulis mampu menyelesaikan penelitian ini beserta laporan akhir tepat pada waktunya. Serta dengan karunia-Nya penulis juga mendapat begitu besar limpahan ilmu dan semangat untuk terus berusaha dan berdoa dalam menjalani proses penelitian yang panjang. Sehingga Nikmat yang begitu besar atas penyelesaian tugas akhir ini penulis dapatkan.

Akhir dari kerja keras penulis, diwujudkan dalam penyusunan dan penulisan laporan tugas akhir penelitian dengan judul **Sistem Pengendali Temperatur dan Kecepatan *Shaking Water Bath* Berbasis *Microcontroller***. Laporan tugas akhir ini menjadi sebuah pengembangan analisa dan pengkajian secara teknis terhadap objek penelitian.

Sebagai bentuk ungkapan terima kasih, penulis memberikan sanjungan serta penghargaan kepada seluruh pihak atas andilnya terhadap penyusunan laporan tugas akhir ini. Pujian penulis secara tulus diberikan dan diucapkan kepada:

1. Allah SWT, Tuhan Yang Majha Esa, yang hingga detik ini senantiasa melimpahkan kasih sayang, rahmat, nikmat, dan ridho-Nya
2. Bapak dan Ibu tercinta untuk doa dan semangatnya tidak pernah berkurang, dan untuk adik-adikku tersayang yang selalu menghilangkan beban dan penat. Serta untuk mbah kung dan mbah uty, yang setia mendoakan dan memotivasi. Juga untuk semua saudara dan kerabat.
3. Dr. Prawito, selaku Ketua Jurusan Program Diploma 3 Instrumentasi Elektronika dan Industri Departemen Fisika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Indonesia, yang dedikasi pengabdian dan jasanya terus tercurah dalam membimbing dan mendidik penulis.
4. Bapak Surya Dharma, M.Si selaku koordinator Tugas Akhir Program Diploma 3 Instrumentasi Elektronika dan Industri Departemen Fisika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Indonesia.

5. Arief Sudarmaji, M.T, selaku dosen dan pembimbing penulis pada Program Diploma 3 Instrumentasi Elektronika dan Industri Departemen Fisika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Indonesia.. Terima kasih atas bimbingan dan kerjasamanya selama proses penelitian sehingga penelitian berjalan baik dan menyenangkan.
6. Seluruh dosen dan staff pengajar Program Diploma 3 Instrumentasi Elektronika dan Industri Departemen Fisika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Indonesia.
7. Pak Katman dan Pak Parno yang selalu membantu dan memudahkan dalam peminjaman alat selama penelitian di Laboratorium Elektronika dan Bengkel Mekanik.
8. Sulastriani, sebagai parner kerja hingga penelitian ini selesai.
9. Desti, Dira, Ervan dan Rosita, yang senantiasa bekerja sama selama penelitian. Serta tim hura, Firman, Turnip, Cungkring, dan Shaddiq yang setia bergadang bersama dengan banyak makanan. Begitu pula untuk temen-teman yang bersama-sama berjuang menyelesaikan tugas akhir semester ini.
10. Semua teman-teman Instrumentasi05 yang senantiasa berjuang bersama dan saling membantu, khususnya untuk Dila, Uli, dkk (*Jilbaber's*) sebagai tim konsumsi dan *Joglo's*.
11. Ka Cahyo, Ka Agung, Ka Usman, dan Ka Yudith yang setia membantu dan menyemangati walaupun dengan keterbatasan waktu dan tempat.
12. Ka Rani, Ka Podang, Ka Arif, Ka Seno, Ka Dody yang juga banyak membantu dan meminjamkan peralatan kerja dalam proses penelitian.
13. Kakak-kakak dan Adik-adik Instumentasi yang selalu menyemangati dan mendoakan.
14. Dan seluruh pihak yang tidak bisa saya sebutkan satu persatu

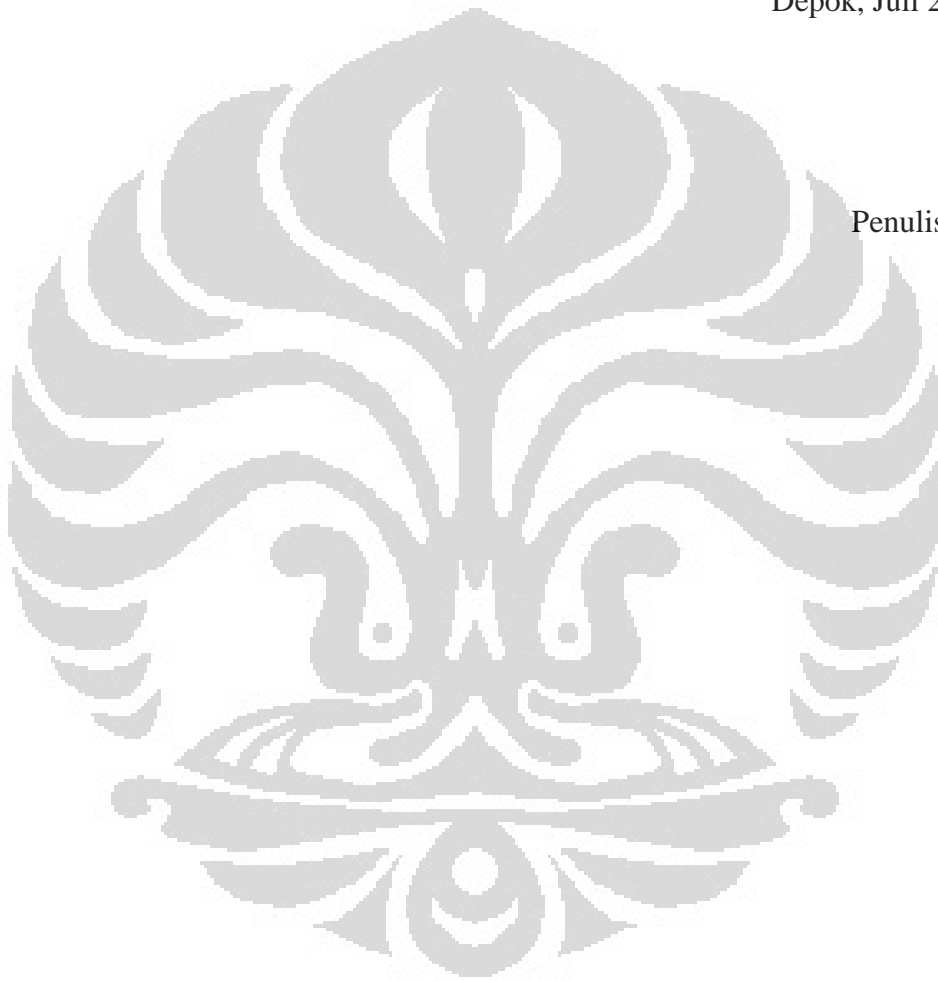
Penulis menyadari bahwa laporan ini banyak kekurangannya dan penulis terbuka untuk menerima kritik dan saran yang bersifat membangun untuk penyusunan laporan agar tercipta peningkatan mutu dan kualitas penulisan lapotan tugas akhir yang lebih baik.

Diharapkan Laporan Tugas Akhir ini bermanfaat bagi pembaca untuk menjadi panduan di kalangan para akademisi dan seluruh civitas akademika Universitas Indonesia, dan bagi penulis pada khususnya.

Universitas Indonesia

Depok, Juli 2008

Penulis



## ABSTRAK

Telah dibuat sebuah alat kontrol yang digunakan untuk mempercepat reaksi suatu larutan dengan cara mengendalikan temperatur dan kecepatan putaran keranjang pada *Shaking Water Bath*. Pengaturan kecepatan putaran pengaduknya dengan cara mengatur kecepatan pada motor DC dan sensor temperaturnya menggunakan termokopel tipe K. Alat ini dilengkapi dengan *heater* sebagai pemanas untuk menaikkan temperatur larutan. Digunakan sebuah keypad atau komputer (PC) untuk mengendalikan dan menampilkan nilai dari kecepatan putaran pengaduk (rpm) dan temperatur ( $^{\circ}\text{C}$ ) dengan memasukkan nilai *Set Point* (SP).

*Kata kunci— Motor DC, Termokopel, pengendali*

## ABSTRACT

A controller which function is to enhance the solution's reaction by controlling the temperature and the basket rotation of the Shaking Water Bath has been build. Controlling the rotation is done by setting the speed on motor DC and temperature sensor using termocouple type K. This machine is equipped with heater to increase solution temperature. A keypad or computer (PC) to control and display the value from the rotation speed (rpm) and temperature ( $^{\circ}\text{C}$ ) by entering the *Set Point* (SP) value.

*Key words— Motor DC, Termocouple, controller*

## DAFTAR ISI

	Halaman
<b>Halaman Judul</b> .....	<b>i</b>
<b>Lembar Pengesahan</b> .....	<b>ii</b>
<b>Kata Pengantar</b> .....	<b>iv</b>
<b>Abstrak</b> .....	<b>vii</b>
<b>Daftar Isi</b> .....	<b>viii</b>
<b>Daftar Gambar</b> .....	<b>x</b>
<b>Daftar Tabel</b> .....	<b>xii</b>
<b>BAB I PENDAHULUAN</b> .....	<b>1</b>
1.1 Latar belakang .....	1
1.2 Tujuan Penelitian .....	2
1.3 Pembatasan Masalah .....	2
1.4 Metodologi Penelitian .....	3
1.5 Sistematika Penulisan .....	5
<b>BAB II TEORI DASAR</b> .....	<b>6</b>
2.1. Motor DC .....	6
2.1.1 Prinsip Kerja Motor DC.....	6
2.1.2 Cara Membalik Arah Motor DC.....	8
2.1.3 Cara Mempercepat Putaran Motor DC .....	9
2.2. PWM ( <i>Pulse Width Modulation</i> ) .....	9
2.3 Sensor Termokopel .....	11
2.4 <i>Microcontroller</i> AT89S52 .....	14
2.5 <i>Microcontroller</i> AVR ATmega16 .....	17
2.6 Komunikasi Data Serial .....	20
2.7 Keypad .....	22
2.8 Teori Dasar Pengendalian PID .....	23



<b>BAB 3. PERANCANGAN DAN CARA KERJA SISTEM .....</b>	<b>28</b>
3.1. Perancangan Kerja Alat. ....	28
3.2 Perancangan Perangkat Keras (Hardware) .....	30
3.2.1. Rangkaian <i>Power Supply</i> .....	30
3.2.2. Rangkaian <i>Minimum System</i> Atmega 16 .....	31
3.2.3. Rangkaian Pengendali Motor DC .....	32
3.2.4. Rangkaian <i>Cold Junction</i> .....	33
3.2.5 Rangkaian Keypad .....	34
3.3 Perancangan Perangkat Lunak (Software) .....	35
3.3.1. Flowchart Proses pada Rangkaian Keypad .....	35
3.3.2. Flowchart Program Pada Sistem Minimum Atmega16 .....	37
<b>BAB 4. HASIL PERCOBAAN DAN ANALISA.....</b>	<b>39</b>
4.1. Pengujian Termokopel .....	39
4.2. Pengujian Motor .....	43
4.3. Pengujian Sistem .....	43
<b>BAB V PENUTUP .....</b>	<b>48</b>
5.1 Kesimpulan .....	48
5.2 Saran .....	49
<b>Daftar Pustaka</b>	
<b>Lampiran A</b>	
<b>Lampiran B</b>	

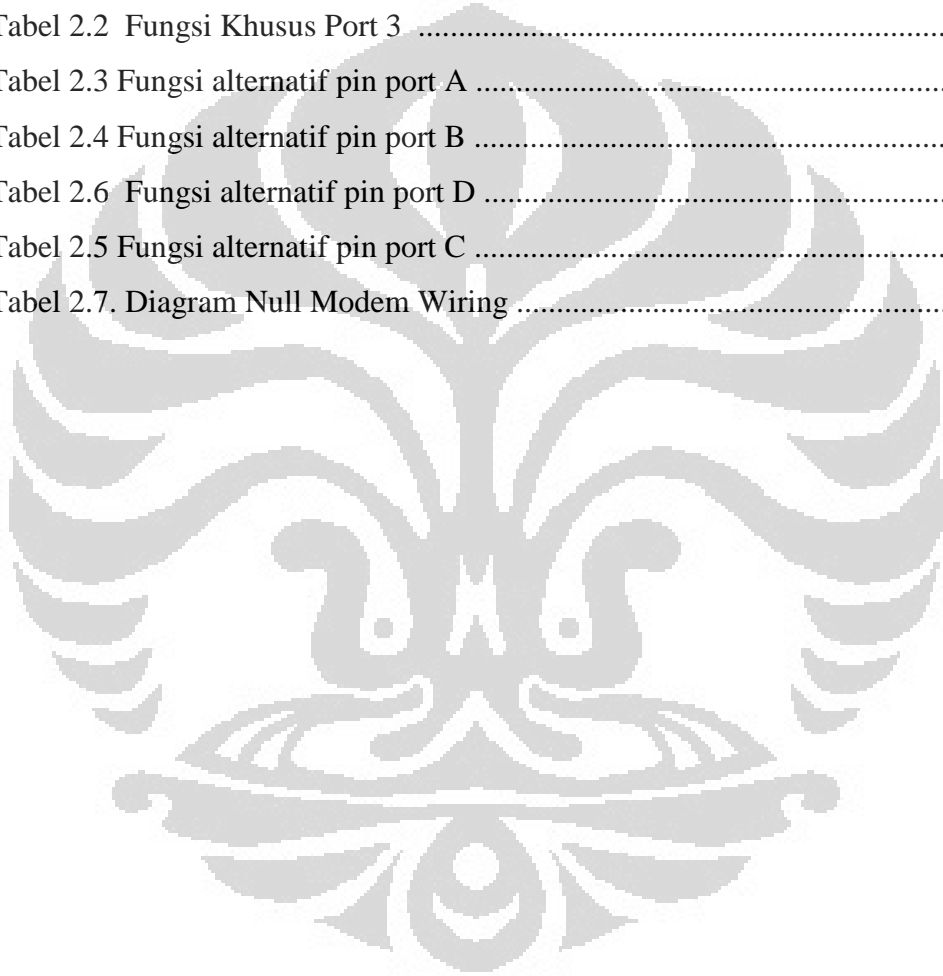
## DAFTAR GAMBAR

	Halaman
Gambar 1.1 Bagan Keseluruhan Sistem .....	3
Gambar 1. 2 Skema Penelitian .....	4
Gambar 2.1 Menentukan arah Gaya Magnet dengan kaidah tangan kanan .....	7
Gambar 2.2. Posisi awal gerakan motor .....	8
Gambar 2.3. Posisi motor setelah 180°.....	8
Gambar 2.4. Sinyal PWM dengan duty cycle 50% .....	10
Gambar 2.5. Sinyal PWM dengan <i>duty cycle</i> 10% .....	10
Gambar 2.6. Termokopel berdasarkan prinsip seebeck .....	12
Gambar 2.7. Pin-pin 89S52 .....	17
Gambar 2.8. Konfigurasi Pin ATmega16 .....	18
Gambar 2.9. Urutan Letak Pin Female RS232 DB-9 .....	21
Gambar 2.10. Urutan Letak Pin Male RS232 DB-9 .....	21
Gambar 2.11. Sistem Pengendali <i>Loop</i> Terbuka .....	24
Gambar 2.12 Sistem Pengendali <i>Loop</i> Tertutup .....	24
Gambar 2.13 .Blok Diagram Pengendali <i>Proportional</i> .....	25
Gambar 2.14. Grafik respon Pengendali <i>Proportional</i> .....	25
Gambar 2.15. Blok Diagram Pengendali <i>Integral</i> .....	26
Gambar 2.16. Grafik Respon Pengendali <i>Integral</i> .....1.....	27
Gambar 2.17. Blok Diagram Pengendali <i>Differensial</i> .....	27
Gambar 3.1. Blok diagram cara kerja alat .....	28
Gambar 3.2. Blok Diagram Pengendali Kecepatan Motor DC .....	29

Gambar 3.3. Blok Diagram Pengendali Temperatur .....	29
Gambar 3.4. Rangkaian <i>Power Supply</i> .....	31
Gambar 3.5. Minimum System <i>Microcontroller</i> AVR .....	31
Gambar 3.6. Rangkaian <i>Driver</i> Motor DC .....	32
Gambar 3.7. Rangkaian <i>Cold Junction</i> .....	33
Gambar 3.8. Rangkaian <i>Keypad</i> .....	34
Gambar 3.9. <i>Flowchart</i> program Keypad .....	36
Gambar 3.10. <i>Flowchart</i> program Pengendali .....	38
Gambar 4.1. LabVIEW untuk pengambilan data ADC .....	39
Gambar 4.2. Grafik Persamaan ADC.....	40
Gambar 4.3. Grafik perbandingan antara nilai ADC <i>hyperterminal</i> pada komputer dengan temperatur pada termometer .....	41
Gambar 4.4. Respon waktu proses perubahan temperatur dengan daya 10% .....	42
Gambar 4.5. Respon waktu proses perubahan temperatur dengan daya 90% .....	42
Gambar 4.6. Kurva Reaksi Dari Respon Sistem .....	44
Gambar 4.7. Grafik persamaan PID dengan setpoint 50 °C.....	45
Gambar 4.8. Grafik persamaan PI dengan setpoint 30 °C .....	46
Gambar 4.9. Grafik persamaan PI dengan setpoint 80 °C .....	47

## DAFTAR TABEL

	Halaman
Tabel 2.1 Perbandingan antar-IC Keluarga MCS-51 .....	15
Tabel 2.2 Fungsi Khusus Port 3 .....	16
Tabel 2.3 Fungsi alternatif pin port A .....	18
Tabel 2.4 Fungsi alternatif pin port B .....	18
Tabel 2.6 Fungsi alternatif pin port D .....	19
Tabel 2.5 Fungsi alternatif pin port C .....	19
Tabel 2.7. Diagram Null Modem Wiring .....	21



## **BAB 1**

### **PENDAHULUAN**

Pada Bab I akan dibahas mengenai Pendahuluan yaitu tentang latar belakang dibuatnya Tugas Akhir ini. Selain itu penulis juga akan membahas tujuan penulisan, pembatasan masalah, metode penelitian yang disertai skema penelitian, dan sistematika penulisan yaitu digunakan dalam penulisan ini dengan membagi menjadi beberapa bab dan masing-masing terdiri dari sub bab.

#### **1.1 LATAR BELAKANG**

Semakin meningkatnya kemajuan ilmu pengetahuan dan teknologi, diikuti dengan meningkatnya sumber daya manusianya. Hal tersebut mempengaruhi kehidupan manusia. Pengaruhnya menjangkau segala bidang, dimana pengerjaan suatu proses mempunyai tuntutan untuk cepat dan tepat. Sehingga banyak dilakukan penelitian untuk menunjang kebutuhan manusia akan perkembangan ilmu pengetahuan dan teknologi

Proses pencampuran dan pemanasan larutan pada bidang kesehatan dan penelitian sering kali menghadapkan kita pada berbagai resiko, diantaranya temperatur yang sangat tinggi dan jenis bahan berbahaya yang digunakan. Pencampuran dan pemanasan tersebut juga memiliki berbagai faktor agar pencampuran dan pemanasan berlangsung aman, efisien dan proses berjalan sempurna. Dalam kenyataannya pengendalian temperatur dan kecepatan gerak pencampuran sangat penting dalam pembuatan alat ini, karena proses pencampuran dan pemanasan berjalan pada temperatur dan kecepatan yang stabil. Pada proses penelitian, alat pencampur dan pemanas biasanya membutuhkan temperatur yang sangat tinggi, sehingga membutuhkan daya listrik yang besar dan waktu yang lama. Akan tetapi dalam beberapa penelitian hanya membutuhkan temperatur yang tidak begitu tinggi dan daya listrik yang kecil agar dapat dikerjakan dalam kapasitas daya rumah tangga.

Dengan latar belakang diatas, saya ingin membuat alat yang mampu menghapus kekurangan alat yang telah ada sebelumnya. Untuk itu saya akan membuat alat pencampur dan pemanas yang terkendali yaitu Sistem Pengendali Temperatur dan Kecepatan Shaking Water Bath Berbasis *Microcontroller*.

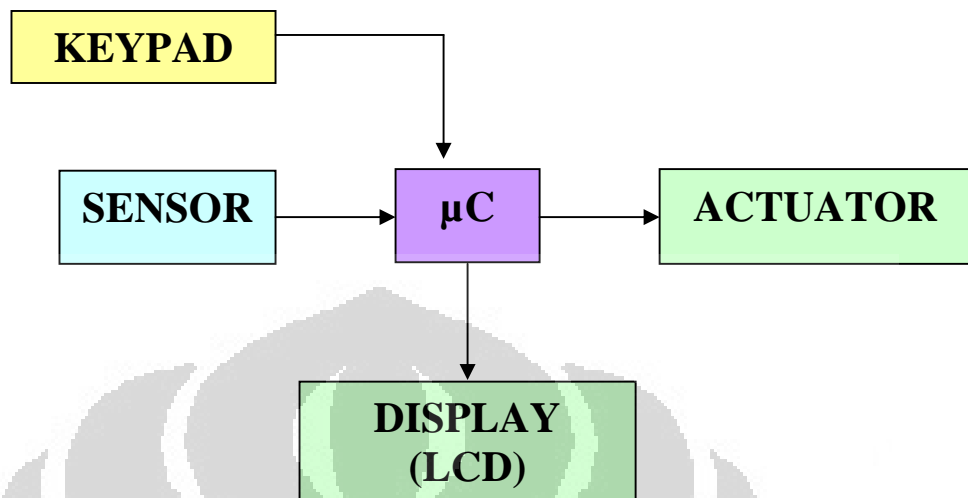
*Shaking Water Bath* dibuat secara mekanik dengan rangkaian elektronik yang dapat menjalankan benda uji, dengan pengendalian temperatur dan kecepatan. Penelitian dilakukan dengan input data pada *keypad* dan memproses data pengendalian menggunakan *Microcontroller* yang dikirim ke rangkaian elektronika kemudian menjalankan sistem mekanik.

## **1. 2. TUJUAN PENELITIAN**

1. Membuat sistem pengendali temperatur dan kecepatan pada *Shaking Water Bath*.
2. Menerapkan dan mengembangkan sistem pengendali menggunakan *Mikrokontroler* sebagai pengendali temperatur dan kecepatan, serta pemroses data sehingga sistem mekanik dapat berjalan.
3. Agar mahasiswa dapat mengetahui, dan mampu mengaplikasikan terhadap ilmu pengetahuan dan teori yang didapatkan dibangku perkuliahan dengan kondisi sebenarnya yang terjadi di lapangan, sehingga hasil dari aktivitas tersebut dapat membangkitkan mahasiswa lebih banyak membuka wawasan dalam bertindak atau berinovasi untuk pengembangan ilmu pengetahuan baik secara teori maupun praktek.

## **1. 3. PEMBATASAN MASALAH**

Dalam pembuatan Tugas Akhir ini terdiri dari proses pembuatan *hardware* dan *software*. Pada pembuatan *hardware* meliputi sistem mekanik pada *Shaking Water Bath*. Sedangkan pada pembuatan *software* meliputi program yang digunakan untuk mempermudah *user* dalam menjalankan mekanik yang kendalikan *Microcontroller* dan di kontrol menggunakan *keypad*. Dan penulis akan membahas mengenai cara kerja program dalam mengatur sistem pengendali temperatur dan kecepatan. Berikut ini adalah bagan dari keseluruhan sistem:



Sistem Pengendali Temperatur dan Kecepatan *Shaking Water Bath*

**Gambar 1.1 Bagan Keseluruhan Sistem**

#### **I. 4. METODOLOGI PENELITIAN**

Metode penelitian yang akan dilakukan terdiri dari beberapa tahap antara lain:

##### **1. Studi Literatur**

Metode ini digunakan untuk memperoleh informasi tentang teori-teori dasar sebagai sumber penulisan skripsi. Informasi dan pustaka yang berkaitan dengan masalah ini diperoleh dari literatur, penjelasan yang diberikan dosen pembimbing, rekan-rekan mahasiswa, internet, *datasheet*, dan buku-buku yang berhubungan dengan tugas akhir penulis.

##### **2. Perancangan dan Pembuatan Alat**

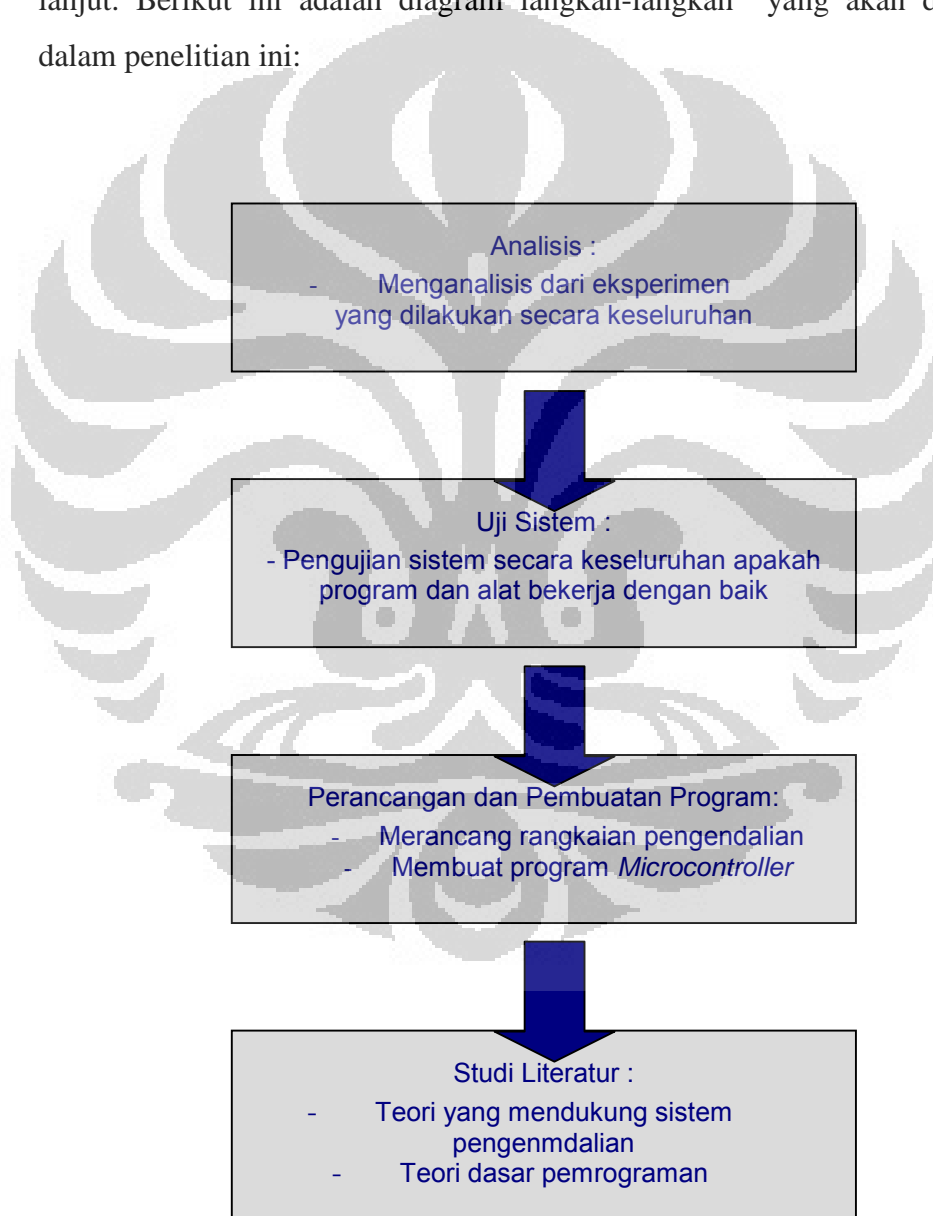
Perancangan alat merupakan tahap awal penulis untuk mencoba memahami, menerapkan, dan menggabungkan semua literatur yang diperoleh maupun yang telah dipelajari untuk melengkapi sistem serupa yang pernah dikembangkan, dan selanjutnya penulis dapat merealisasikan sistem sesuai dengan tujuan, khususnya mengenai sistem pengendalian menggunakan *Microcontroller*.

### 3. Uji Sistem

Uji sistem ini berkaitan dengan pengujian alat dan program serta pengambilan data.

### 4. Metode Analisis

Metode ini merupakan pengamatan terhadap data yang diperoleh dari pengujian alat serta pengambilan data. Setelah itu dilakukan penganalisaan sehingga dapat ditarik kesimpulan dan saran-saran untuk pengembangan lebih lanjut. Berikut ini adalah diagram langkah-langkah yang akan dilakukan dalam penelitian ini:



**Gambar 1. 2 Skema Penelitian**



## **1. 5. 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 I Pendahuluan**

Pendahuluan berisi latar belakang, permasalahan, batasan masalah, tujuan penulisan, metode penulisan dan sistematika penulisan dari tugas akhir ini.

### **BAB II Teori Dasar**

Berisi teori yang mendasari penelitian, yaitu prinsip dasar kerja program sebagai pengendali utama temperatur dan kecepatan.

### **BAB III Perancangan Sistem**

Pada bab ini akan dijelaskan secara keseluruhan sistem kerja dari semua perangkat kontrol ( *hardware* ) dan program penghubung ( *software* ) yang terlibat antara mikrokontroler dengan rangkaian elektronik dan *keypad*.

### **BAB IV Pengujian Sistem dan Pengambilan Data**

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 V Penutup**

Berisi kesimpulan dari keseluruhan perancangan sistem hingga hasil penelitian yang didapat dan saran yang mungkin dapat digunakan untuk memperbaiki, menambahkan, ataupun memodifikasi alat yang sudah ada menjadi lebih baik lagi.

## BAB 2

### TEORI DASAR

Pada perancangan alat pengendali temperatur dan kecepatan pada Shaking Water Bath, diperlukan beberapa pemahaman dasar. Beberapa pemahaman dasar tersebut antara lain: prinsip dasar motor DC, PWM (*Pulse Width Modulation*), Sensor Temperatur, Pengendali, *Keypad*, dan *Display* berupa LCD.

#### 2.1. Motor DC

Motor DC adalah motor yang menggunakan arus searah (*Direct Current*) dan ini dapat mengubah energi listrik menjadi energi mekanik berupa gerak berputar. Salah satu komponen yang diperlukan dalam sistem pengendali adalah aktuator, yaitu komponen pertama untuk melakukan gerakan dengan mengubah energi elektrik menjadi gerakan mekanik dan salah satu jenis aktuator adalah motor listrik.

##### 2.1.1. Prinsip Kerja Motor DC

Prinsip kerja motor DC dapat dijelaskan dengan teori *elektromagnetik*. Misal sebuah kawat berarus yang dipengaruhi medan magnet luar akan mengalami gaya yang disebut gaya magnet yang besarnya ditunjukkan pada persamaan:

$$F = BiL \sin \alpha \dots\dots\dots(2.1)$$

Dimana:

F = gaya magnet (Newton)

B = medan magnet luar (Wb/m<sup>2</sup>)

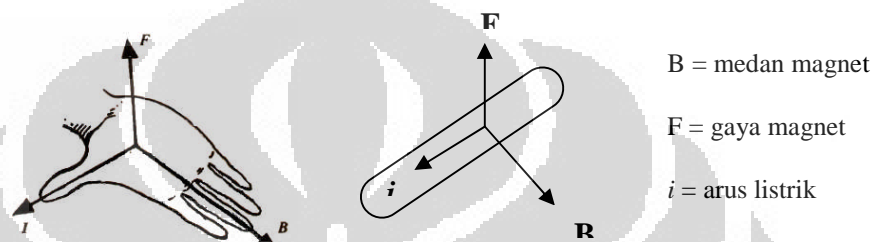
i = kuat arus (Ampere)

L = panjang kawat (Meter)

$\alpha$  = sudut yang dibentuk medan magnetik dengan arus

Pada gambar 2.1 sebuah kawat berarus listrik di dalam pengaruh medan magnet sehingga kawat tersebut memiliki gaya magnet. Arah dari gaya magnet (F) dapat ditentukan dengan kaidah tangan kanan, yaitu :

” Keempat jari yang dirapatkan arah induksi magnetik B dan arus listrik I mengalir searah dengan ibu jari, maka kawat itu akan mendapatkan gaya magnetik F adalah searah dengan arah dorong telapak tangan”.

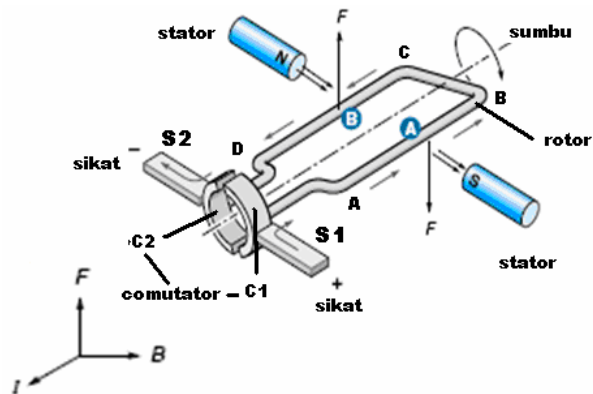


**Gambar 2.1 Menentukan arah Gaya Magnet dengan kaidah tangan kanan**

Motor DC terdiri dari bagian-bagian yang dapat menggerakkan motor tersebut, yaitu:

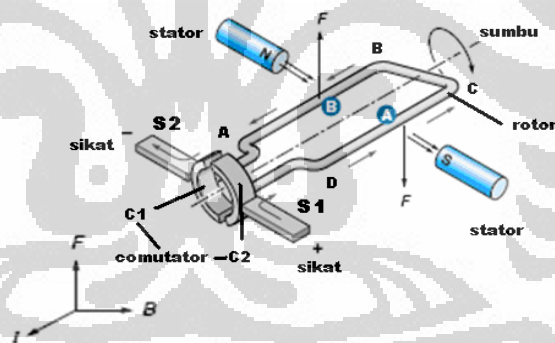
1. **Stator**, merupakan bagian yang diam pada motor berupa magnet.
2. **Rotor**, merupakan bagian yang berputar pada motor berupa kumparan kawat.
3. **Komutator**, merupakan cincin belah yang berfungsi sebagai penukar arus.
4. **Sikat**, merupakan sepasang batang grafit yang menempel pada komutator tetapi tidak berputar.

Cara kerja motor DC dapat dilihat pada gambar 2.2, misalkan kedudukan mula-mula seperti pada gambar 2.2 arus listrik mengalir dari kutub (+) baterai melalui sikat S1 – cincin C1 - rotor ABCD – cincin C2 – sikat S2 – kembali ke kutub (-) baterai. Ketika rotor CD yang dekat dengan kutub utara mengalami gaya ke atas dan sisi rotor AB yang dekat dengan kutub selatan mengalami gaya ke bawah. Akibatnya rotor ABCD berputar searah jarum jam.



**Gambar 2.2. Posisi awal gerakan motor**

Setelah setengah putaran ( $180^\circ$ ), terjadi pertukaran posisi antara sikat dan komutator (gambar 2.3). Sekarang, C2 menyentuh sikat S1 dan C1 menyentuh sikat S2. Sehingga arus mengalir dari kutub (+) baterai menuju kutub (-) melalui sikat 1 (S1), Komutator 2 (C2), Rotor DCBA, Komutator 2 (C2), dan sikat 2 (S2). Pertukaran posisi antara sikat dan Komutator mengakibatkan motor terus berputar.



**Gambar 2.3. Posisi motor setelah  $180^\circ$**

### 2.1.2. Cara Membalik Arah Motor DC

Arah gerakan motor arus searah dapat diatur dengan dua cara yaitu mengubah polarisasi arah arus searah pada belitan medan magnet (+) dan (-), atau dengan mengubah arah arus dengan menukar (+) dan (-) pada sikat.

Pada prinsipnya membalik arah motor searah memang dengan dua cara yang telah disebutkan di atas, namun dalam suatu rangkaian elektronika kita memerlukan suatu rangkaian penggerak motor yang dapat membalik arah gerak

motor dengan mudah misalnya dengan menggunakan transistor. Transistor pada rangkaian pembalik putaran motor berfungsi sebagai saklar (*switching*).

### 2.1.3. Cara Mempercepat Putaran Motor DC

Kecepatan putaran motor DC dapat ditingkatkan dengan memperbesar tegangan yang masuk ke motor, sehingga dapat mengakibatkan arus yang masuk ke motor menjadi besar pula. Hal ini sesuai dengan hukum Ohm berikut ini:

$$V = i \times R \dots\dots\dots(2.2)$$

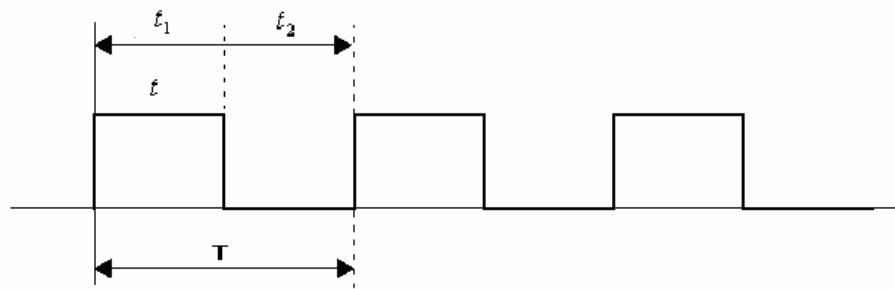
Dimana:

- V = Tegangan (Volt)
- i = Besar arus (Ampere)
- R = Hambatan (Ohm)

Dengan hambatan yang tetap dan tegangan diperbesar akan mengakibatkan arus menjadi besar pula. Dengan arus yang diperbesar maka akan menyebabkan gaya ( $F$ ) menjadi besar pula sesuai dengan persamaan 2.1 di atas. Dan apabila  $F$  semakin besar maka kekuatan rotor akan semakin besar dan berdampak pada makin cepatnya putaran motor. Dari persamaan 2.1 dapat diambil kesimpulan bahwa semua unsur yang mempengaruhi nilai  $F$  dapat mempercepat putaran motor, yaitu dengan memperpanjang lilitan (memperbesar  $I$ ), dan memperbesar medan magnet ( $B$ ).

### 2.2. 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 sebuah 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 2.4 memperlihatkan sinyal kotak dengan *duty cycle* 50%.



**Gambar 2.4. Sinyal PWM dengan duty cycle 50%**

*Duty cycle* adalah rasio dari waktu ON ( $t_{On}$ ) terhadap periode total dari sinyal ( $t = t_{On} + t_{Off}$ ). Dengan persamaan :

$$D = \frac{t_1}{t_1 + t_2} \dots\dots\dots(2.3)$$

Dimana :  $t_1$  = Pulsa ON

$t_2$  = Pulsa OFF

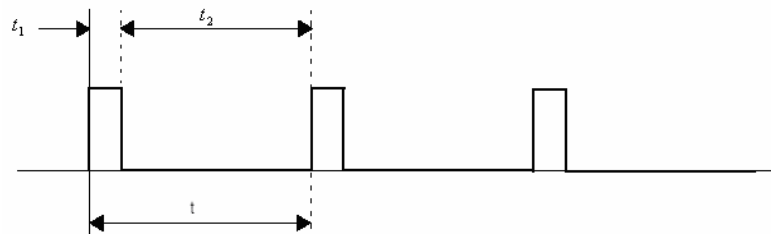
$t_2$  = Pulsa OFF

Sehingga frekuensi yang dapat dihasilkan :

$$F = \frac{1}{T} \dots\dots\dots(2.4)$$

Dimana :  $T$  = periode ( $t_1 + t_2$ )

Dengan *duty cycle* yang bermacam-macam, rata-rata output dari tegangan DC dapat dikontrol. Seperti pada gambar 2.5 memperlihatkan sinyal kotak dengan *duty cycle* 10%.



**Gambar 2.5. Sinyal PWM dengan duty cycle 10%**

Dari gambar diatas maka kita dapat mengetahui hubungan daya dengan waktu dimana :

$$V_{rms} = \sqrt{\frac{\int_0^t V_{(t)}^2 dt}{T}} \dots\dots\dots [4] (2.5)$$

Dengan :

$$V_{(t)} \text{ pada } 0 - t ; V_{(t)} = V_{(p)} \text{ dan } V_{(t)} \text{ pada } t - T ; V_{(t)} = 0$$

Maka :

$$V_{rms} = \sqrt{\frac{\int_0^t V_p^2 dt}{T}} \dots\dots\dots [4](2.6)$$

$$V_{rms} = \sqrt{\frac{V_p^2 t}{T}} \dots\dots\dots [4](2.7)$$

$$V_{rms} = V_p \sqrt{\frac{t}{T}} \dots\dots\dots [4](2.8)$$

$$Power = \frac{V_{rms}^2}{R} \dots\dots\dots [4](2.9)$$

$$Power = \frac{V_p^2 (t)}{R (T)} \dots\dots\dots [4](2.10)$$

### 2.3. Sensor Termokopel

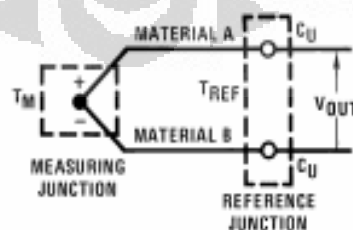
Sensor secara umum didefinisikan sebagai alat yang mampu menangkap fenomena fisik atau kimia kemudian mengubahnya menjadi sinyal elektrik baik arus listrik maupun tegangan. Fenomena fisik yang mampu stimulasi sensor untuk menghasilkan sinyal elektrik meliputi temperatur, tekanan, gaya, dan sebagainya. Sensor itu sendiri terdiri dari *transducer* dengan atau tanpa penguat/pengolah sinyal yang berbentuk dalam suatu sistem penginderaan.

Temperatur adalah satu dari besaran fisika yang terpenting. Alat yang dapat merubah besaran fisis kesuatu besaran yang bersifat listrik. Temperatur didefinisikan sebagai ukuran relatif dari kondisi termal yang dimiliki suatu benda. Alat pengukur temperatur disebut termometer.

Sensor temperatur adalah suatu *transducer* yang menangkap perubahan temperatur menjadi suatu besaran fisika lain, seperti tegangan atau arus. Sensor temperatur yang digunakan dalam skripsi ini salah satunya adalah termokopel.

Termokopel adalah sensor suhu yang banyak digunakan untuk mengubah perbedaan panas dalam benda yang diukur temperaturnya menjadi perubahan potensial atau tegangan listrik (*voltage*). Termokopel 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.

Berdasarkan prinsip *seebeck* termokopel merupakan sensor temperatur termoelektrik yang terdiri dari sepasang kawat logam konduktor berbeda jenis dihubungkan bersama-sama pada satu ujung (ujung pengindera atau ujung panas) sebagai pengukur dan berakhir pada ujung lain (titik referensi) yang dipertahankan pada suatu temperatur konstan yang diketahui temperatur referensinya. Bila antara ujung penginderaan dan titik referensi terdapat perbedaan temperatur maka akan dihasilkan suatu tegangan. Sangat penting diingat bahwa termokopel mengukur perbedaan temperatur di antara 2 titik, bukan temperatur absolut.



**Gambar 2.6. Termokopel berdasarkan prinsip seebeck**



Hubungan antara perbedaan suhu dengan tegangan yang dihasilkan termokopel bukan merupakan fungsi linier melainkan fungsi interpolasi polinomial. Koefisien  $n$  memiliki  $n$  antara 5 dan 9. Agar diperoleh hasil pengukuran yang akurat, persamaan biasanya diimplementasikan pada kontroler digital atau disimpan dalam sebuah tabel pengamatan. Beberapa peralatan yang lebih tua menggunakan filter analog.

Tersedia beberapa jenis termokopel tergantung aplikasi penggunaannya, yaitu:

- Tipe K (Chromel (Ni-Cr alloy) / Alumel (Ni-Al alloy)) Termokopel untuk tujuan umum. Lebih murah. Tersedia untuk rentang suhu  $-200\text{ }^{\circ}\text{C}$  hingga  $+1200\text{ }^{\circ}\text{C}$ .
- Tipe E (Chromel / Constantan (Cu-Ni alloy))
- Tipe E memiliki output yang besar ( $68\text{ }\mu\text{V}/^{\circ}\text{C}$ ) membuatnya cocok digunakan pada temperatur rendah. Properti lainnya tipe E adalah tipe non magnetik.
- Tipe J (Iron / Constantan) Rentangnya terbatas ( $-40$  hingga  $+750\text{ }^{\circ}\text{C}$ ) membuatnya kurang populer dibanding tipe K
- Tipe J memiliki sensitivitas sekitar  $\sim 52\text{ }\mu\text{V}/^{\circ}\text{C}$
- 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  $1200\text{ }^{\circ}\text{C}$ . Sensitifitasnya sekitar  $39\text{ }\mu\text{V}/^{\circ}\text{C}$  pada  $900^{\circ}\text{C}$ , sedikit di bawah tipe K. Tipe N merupakan perbaikan tipe K
- Termokopel tipe B, R, dan S adalah termokopel logam mulia yang memiliki karakteristik yang hampir sama. Mereka adalah termokopel yang paling stabil, tetapi karena sensitifitasnya rendah (sekitar  $10\text{ }\mu\text{V}/^{\circ}\text{C}$ ) mereka biasanya hanya digunakan untuk mengukur temperatur tinggi ( $>300\text{ }^{\circ}\text{C}$ ).
- Tipe B (Platinum-Rhodium/Pt-Rh) Cocok mengukur suhu di atas  $1800\text{ }^{\circ}\text{C}$ . Tipe B memberi output yang sama pada suhu  $0^{\circ}\text{C}$  hingga  $42^{\circ}\text{C}$  sehingga tidak dapat dipakai di bawah suhu  $50^{\circ}\text{C}$ .

- Type R (Platinum /Platinum with 7% Rhodium) Cocok mengukur suhu di atas 1600 °C. sensitivitas rendah ( $10 \mu\text{V}/^\circ\text{C}$ ) dan biaya tinggi membuat mereka tidak cocok dipakai untuk tujuan umum.
- Type S (Platinum /Platinum with 10% Rhodium) Cocok mengukur suhu di atas 1600 °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 ( $1064.43 \text{ }^\circ\text{C}$ ).
- Type T (Copper / Constantan) Cocok untuk pengukuran antara  $-200$  to  $350 \text{ }^\circ\text{C}$ . Konduktor positif terbuat dari tembaga, dan yang negatif terbuat dari constantan. Sering dipakai sebagai alat pengukur alternatif sejak penelitian kawat tembaga. Type T memiliki sensitifitas  $\sim 43 \mu\text{V}/^\circ\text{C}$

Untuk pembuatan alat ini menggunakan termokopel tipe K, dimana termokopel tipe K ini memiliki karakteristik tersendiri yaitu :

- Terbuat dari bahan chromel ( Ni-Cr ) dan alumel ( Ni-Al )
- Bekerja pada suhu  $-200^\circ\text{C} \sim 1200^\circ\text{C}$
- Sensitivity @  $25^\circ\text{C}$  adalah  $40,6 \mu\text{V}/^\circ\text{C}$
- Kesalahan dalam bekerja pada :
  - Temperatur rendah  $\pm 2,2-1,1^\circ\text{C}$  (  $\pm 4 \sim 2^\circ\text{F}$  )
  - Temperatur tinggi  $\pm 0,375-0,75\%$

#### 2. 4. Mikrokontroler AT89S52

Mikrokontroler dengan arsitektur AT89S52 merupakan salah satu jenis IC mikrokontroler yang paling lama dan paling banyak digunakan di dunia. Jenis ini dikeluarkan pertama kali oleh Intel dan kemudian menjadi sangat populer. Berbagai seri mikrokontroler berjenis AT89S52 telah diproduksi oleh berbagai perusahaan dan digunakan di dunia sebagai mikrokontroler yang bersifat *low cost* dan *high performance*. Beberapa pabrikan yang terkenal antara lain Atmel, Philips, dan Siemens. Pada Intel, seri-seri mikrokontroler berjenis 8052 tergabung dalam satu keluarga mikrokontroler yaitu keluarga MCS-51.

AT89S52 merupakan single chip computer yang memiliki kemampuan untuk diprogram dan digunakan untuk tugas-tugas yang bertujuan mengontrol. AT89S52 merupakan IC yang digunakan untuk mengolah data (kontrol) pada sistem minimum. AT89S52 memiliki 4 buah port parallel. P0, P1, P2, P3 semuanya bisa sebagai port I/O. P0 membutuhkan *pull up eksternal* selama proses verifikasi program. Seri mikrokontroler berarsitektur 8052, baik dari keluarga Intel MCS-51 maupun dari buatan yang lain. 8052 memiliki beragam tipe dan fasilitas, namun kesemuanya memiliki karakteristik yang sama, dan juga set instruksi yang relatif tidak berbeda dan dapat dilihat pada gambar Diagram Blok MCS-51.

Perbedaan antara seri yang satu dengan yang lain pada keluarga MCS-51 dapat dilihat pada tabel 2.1

**Tabel 2.1 Perbandingan antar-IC Keluarga MCS-51 [5]**

<b>PART NUMBER</b>	<b>ON-CHIP CODE MEMORY</b>	<b>ON-CHIP DATA MEMORY</b>	<b>TIMERS</b>
8051	4 KB ROM	128 bytes	2
8031	0	128 bytes	2
8751	4 KB EPROM	128 bytes	2
8052	8 KB ROM	128 bytes	3
8032	0	128 bytes	3
8752	8 KB EPROM	128 bytes	3
8951	4 KB EEPROM	128 bytes	2

#### **Fungsi Pin Mikrokontroler AT89S52 :**

IC mikrokontroler dikemas (*packaging*) dalam bentuk yang berbeda. Namun pada dasarnya fungsi kaki yang ada pada IC memiliki persamaan. Gambar salah satu bentuk IC seri mikrokontroler AT80S52 dapat dilihat pada gambar 2.16.

Berikut adalah penjelasan fungsi tiap kaki yang biasa ada pada seri mikrokontroler 8052 gambar pin – pin pada AT89S52.

### **Port 0**

Merupakan dual-purpose port (port yang memiliki dua kegunaan). Pada desain yang minimum (sederhana) digunakan sebagai port I/ O (Input/ Output). Port 0 terdapat pada pin 32-39. Kelebihan pada port 0 adalah memiliki internal pull up.

### **Port 1**

Merupakan port yang hanya berfungsi sebagai port I/ O, kecuali pada IC 8032/ 8052 yang menggunakan P1.0 dan P1.1 sebagai input eksternal untuk timer ketiga (T3). Port 1 terdapat pada pin 1-8.

Port Pin Fungsi Alternatif

P1.5 MOSI ( digunakan untu In System Programming )

P1.6 MISO ( digunakan untu In System Programming )

P1.7 SCK ( digunakan untu In System Programming )

### **Port 2**

Merupakan *dual-purpose* port. Pada desain minimum digunakan sebagai port I/ O. Pada desain lebih lanjut digunakan sebagai *high* byte dari *address*. Port 2 terdapat pada pin 21-28.

### **Port 3**

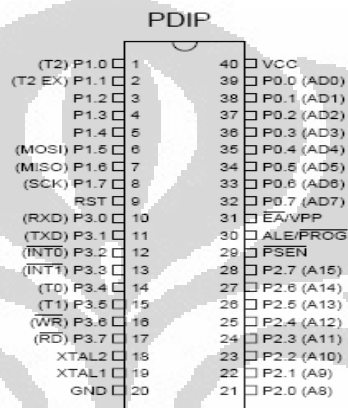
Merupakan *dual-purpose* port. Selain sebagai port I/ O juga mempunyai fungsi khusus yang ditunjukan pada tabel 2. 2.

**Tabel 2.2 Fungsi Khusus Port 3 [5]**

<b>PORT PIN</b>	<b>ALTERNATE FUNCTIONS</b>
P3.0	RXD ( <i>serial input port</i> )
P3.1	TXD ( <i>serial output port</i> )
P3.2	_INT0 ( <i>external interrupt 0</i> )
P3.3	_INT1 ( <i>external interrupt 1</i> )
P3.4	T0 ( <i>timer 0 external input</i> )
P3.5	T1 ( <i>timer 1 external input</i> )
P3.6	_WR ( <i>external data memory write strobe</i> )
P3.7	_RD ( <i>external data memory read strobe</i> )

## On-Chip Oscillator

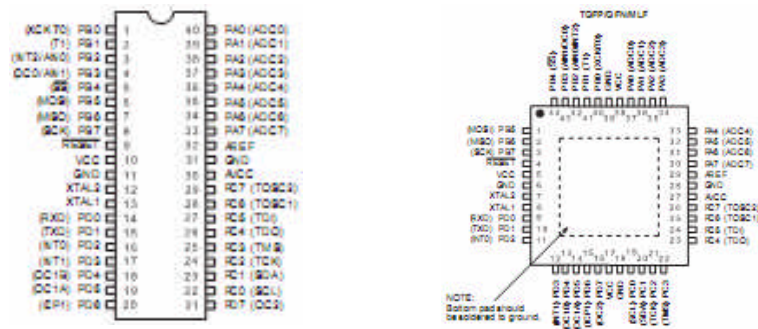
8052 telah memiliki on-chip oscillator yang dapat bekerja jika di-drive menggunakan kristal. Tambahan kapasitor diperlukan untuk menstabilkan system. Nilai kristal yang biasa digunakan pada keluarga MCS-51 adalah 12MHz walaupun pada jenis 80C31BH-1 dapat menggunakan kristal dengan frekuensi sampai 16 Mhz. On-chip oscillator tidak hanya dapat di-drive dengan menggunakan kristal, tapi juga dapat digunakan TTL oscillator.



Gambar 2.7. Pin-pin 89S52

## 2. 5. Mikrokontroler ATmega16

Mikrokontroler ATmega16 merupakan salah satu jenis IC mikrokontroler yang digunakan di dunia Elektronika dengan bahasa pemrograman Basic Compailer Aviar, berbagai seri mikrokontroler berjenis ATmega telah diproduksi oleh berbagai perusahaan dan digunakan di dunia sebagai mikrokontroler yang bersifat *low cost* dan *high performance*, beberapa pabrikan yang terkenal antara lain Atmel, Philips, dan Siemens. ATmega16 merupakan *single chip computer* yang memiliki kemampuan untuk diprogram dan digunakan untuk tugas-tugas yang bertujuan mengontrol. ATmega16 merupakan IC yang digunakan untuk mengolah data (kontrol) pada sistem minimum. ATmega16 memiliki 4 buah port parallel. PA, PB, PC, dan PD semuanya bisa sebagai port I/O. Konfigurasi Pin ATmega16 dapat dilihat pada gambar 8 dibawah ini :



**Gambar 2. 8. Konfigurasi Pin ATmega16**

Konfigurasi Pin Mikrokontroler ATmega16 :

**Port A (PA. 0 – PA. 7)**

Sebagai konverter Input Analog – Digital, fungsi alternatif pin dari port A dapat dilihat pada tabel 2.3

**Tabel 2.3 Fungsi alternatif pin port A [5]**

Pin Port	Fungsi Alternatif
PA 0	A/D Converter Channel 0
PA 1	A/D Converter Channel 1
PA 2	A/D Converter Channel 2
PA 3	A/D Converter Channel 3
PA 4	A/D Converter Channel 4
PA 5	A/D Converter Channel 5
PA 6	A/D Converter Channel 6
PA 7	A/D Converter Channel 7

**Port B (PB. 0 – PB. 7)**

Sebagai Port I/O dengan internal pull-up resistor, fungsi alternatif pin dari port A dapat dilihat pada tabel 2.4

**Tabel 2.4 Fungsi alternatif pin port B [5]**

Pin Port	Fungsi Alternatif
PB 0	T0 (Timer/Counter 0 External counter input)
PB 1	T1 (Timer/Counter 1 External counter input)
PB 2	AIN 0 (input positif analog comparator)
PB 3	AIN 1 (input negatif analog comparator)
PB 4	SS (SPI slave select input)
PB 5	MOSI (SPI bus master output/slave input)
PB 6	MISO (SPI bus master input/slave output)
PB 7	SCK (SPI bus serial clock)

**Port C (PC. 0 – PC. 7)**

Sebagai Port I/O dengan internal pull-up resistor, fungsi alternatif pin dari port A dapat dilihat pada tabel 2.5

**Tabel 2.5 Fungsi alternatif pin port C [5]**

Pin Port	Fungsi Alternatif
PC 0	SCL (two-wire serial bus clock line)
PC 1	SDA (two-wire serial bus data I/O line)
PC 2	TCK (JTAG clock)
PC 3	TMS (JTAG test mode select)
PC 4	TOO (JTAG test data out)
PC 5	TDI (JTAG test data in)
PC 6	TO SC1 (Timer Oscillator Pin1)
PC 7	TO SC2 (Timer Oscillator Pin2)

**Port D (PD. 0- PD. 7)**

Sebagai Port I/O dengan internal pull-up resistor, fungsi alternatif pin dari port A dapat dilihat pada tabel 2.6

**Tabel 2.6 Fungsi alternatif pin port D [5]**

Pin Port	Fungsi Alternatif
PD 0	RXD (USART input pin)
PD 1	TXD (USART output pin)
PD 2	INT0 (External Interrupt 0 input)
PD 3	INT1 (External Interrupt 1 input)
PD 4	OC1B (Timer/Counter1 output compare B match output)
PD 5	OC1A (Timer/Counter1 output compare A match output)
PD 6	ICP1 (Timer/Counter1 input capture pin)
PD 7	OC2 (Timer/Counter2 output compare A match output)

## 2.6. Komunikasi Data Serial

Komunikasi serial adalah pengiriman data secara serial (data dikirim satu persatu secara berurutan) sehingga komunikasi serial jauh lebih lambat daripada komunikasi yang paralel. Karena peralatan berkomunikasi menggunakan transmisi serial sedangkan data dikomputer diolah secara paralel, oleh karena itu harus dikonversikan dahulu ke bentuk paralel. Jika menggunakan perangkat keras hal ini bisa dilakukan oleh *Universal Asynchronous Receiver Transmitter (UART)*, yang membutuhkan perangkat lunak untuk menanganinya. Komunikasi serial merupakan salah satu cara untuk mengkomunikasikan data dari suatu peralatan ke peralatan lain dengan cara menggunakan data secara serial, misalnya mengkomunikasikan antara HP dengan Mikrokontroler, HP dengan PC, *printer* dengan PC dll Pada PC, komunikasi serial RS232 dapat dilakukan melalui *port* serial (COM port).

Komunikasi data serial dapat dilakukan dengan mempresentasikan data dalam bentuk level "1" atau "0" Kelebihan komunikasi serial adalah jangkauan panjang kabel yang lebih jauh dibanding paralel karena serial port mengirimkan logika 1 dengan kisaran tegangan -3 Volt hingga -25 Volt dan logika nol sebagai +3 Volt hingga +25 Volt sehingga kehilangan daya karena panjang kabel bukan masalah utama. Selain itu juga komunikasi serial *port* bersifat asinkron sehingga sinyal detak tidak dikirim bersama data. Setiap *word* disinkronkan dengan *start* bit



dan sebuah *clock internal* di kedua sisi menjaga bagian data saat pewaktuan (*timing*).

Perangkat keras pada komunikasi serial dibagi menjadi dua kelompok, yaitu *Data Communication Equipment* (DCE) dan *Data Terminal Equipment* (DTE). Contoh DCE adalah modem, sedangkan contoh DTE adalah terminal di komputer.



**Gambar 2.9. Urutan Letak Pin Female RS232 DB-9**



**Gambar 2.10. Urutan Letak Pin Male RS232 DB-9**

Null modem digunakan untuk menghubungkan dua buah data DTE . hal ini biasanya dilakukan untuk mentransfer file antar komputer .

**Tabel 2.7. Diagram Null Modem Wiring**

D9	D25				D25	D9
3	2	TD	→	RD	3	2
2	3	RD	←	TD	2	3
5	7	SG	→	SG	7	5
4	20	DTR	←	DTR	20	4
6	6	DSR	←	DSR	6	6
1	8	CD	←	CD	8	1
7	4	RTS	←	RTS	4	7
8	5	CTS	←	CTS	5	8

Komunikasi serial RS 232 merupakan komunikasi *asynchronous* sehingga sinyal *clock* tidak dikirim bersamaan dengan data. Setiap data disinkronisasikan dengan menggunakan *start bit* dan *clock internal* pada setiap bit.

Port RS 232 pada komputer harus memenuhi standar RS 232. Agar level tegangan sesuai dengan tegangan TTL/CMOS diperlukan RS 232 level konverter. IC yang banyak digunakan untuk ini adalah MAX-232.

Jika peralatan yang digunakan melalui RS 232 menggunakan TTL, sinyal serial *port* harus dikonversikan dahulu ke pulsa TTL sebelum digunakan. Sebaliknya, sinyal dari peralatan harus dirubah ke logika RS-232 sebelum dimasukan ke serial *port*. Konverter yang paling mudah digunakan adalah MAX-232. Di dalam IC ini terdapat *Charge Pump* yang akan membangkitkan +10 Volt dari sumber dan +5 Volt tunggal yang dikemas dalam IC DIO (*Dual In Line Package*) 26 pin (8 pin x 2 baris) ini terdapat dua buah *transmitter* dan dua buah *receiver*.

## 2.7. Keypad

Saat pembuatan suatu alat agar mudah dalam pemerintahan kita memerlukan setidaknya suatu komponen yang memberikan data keluaran yang dapat diterima mikrokontroler sebagai input. Seperti *push button*, *limit switch*, sensor, *keypad*, dan sebagainya. Dalam hal ini saya menggunakan *keypad* untuk memberikan data perintah untuk mikrokontroler sebagai penentuan langkah-langkah yang diinginkan.

*Keypad* memiliki berbagai jenis tipe, ada *single key*, matrix 3 X 4, 4 X 4, dan sebagainya.. Dari jenis-jenis *keypad* yang ada memiliki karakteristik tersendiri. Karakter *keypad* menentukan data-data yang dikeluarkan, dari masing-masing *keypad* memiliki nilai keluaran data yang berbeda. Data-data yang dihasilkan akan mempengaruhi berkerja atau tidaknya sebuah *keypad*. Karena bekerja atau tidaknya *keypad* akan mempengaruhi berjalannya suatu program. Jika data *keypad* yang dihasilkan tidak sesuai dengan program maka program tidak dapat berkerja. *Keypad* yang dipakai penulis memiliki 18 buah push button yang mempunyai fungsi berbeda – beda.

*Keypad* merupakan penggabungan *push button* yang dirangkai menjadi sebuah saklar yang mampu memberikan nilai output. *Keypad* dalam kinerjanya seperti saklar yang memiliki hubungan nilai hambatan, ketika salah satu tombol *keypad* tertekan maka akan ada dua hambatan yang akan terhubung. Dalam

pencaraian data kita perlu menghubungkan bit-bit yang ada pada *keypad*, saat ada satu tombol *keypad* yang ditekan maka akan ada dua hambatan yang terhubung. Saat dua hambatan yang terhubung maka *keypad* menghasilkan nilai output yang dapat diterima mikrokontroller. Saat data diterima mikrokontroller maka programakan bekerja sesuai dengan yang diperintahkan.

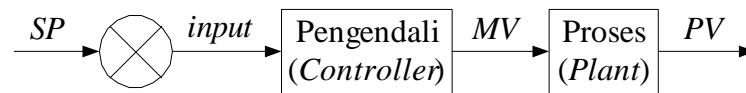
## 2.8. Teori Dasar Pengendali *P.I.D*

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*. 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).

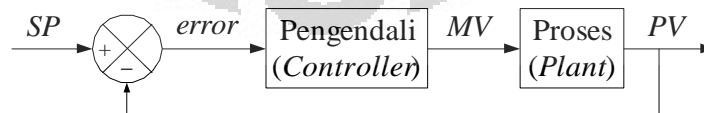
*Open Loop Control System* atau sistem pengendali *loop* terbuka merupakan sistem pengendalian dimana objek yang dikontrol tidak di-*feedback* ke pengendali, sehingga pengendali hanya akan memberikan output jika diberikan suatu sinyal input. Pengendali jenis ini masih bersifat manual karena tidak akan 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 2.7 menggambarkan sistem pengendali *loop* terbuka (*Open Loop Control System*).



**Gambar 2.11. Sistem Pengendali *Loop* Terbuka [3]**

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 2.4 menggambarkan sistem pengendali *loop* tertutup (*Closed Loop Control System*).

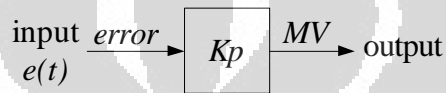


**Gambar 2.12 Sistem Pengendali *Loop* Tertutup [3]**

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.

**a. Pengendali *Proportional* (P)**

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



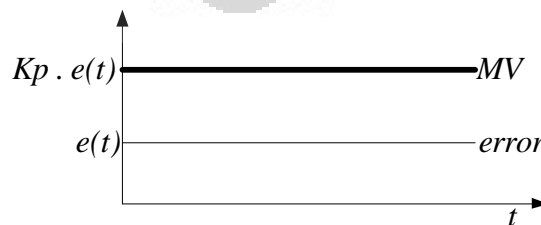
**Gambar 2.13 .Blok Diagram Pengendali *Proportional* [3]**

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

$$\text{output} = K_p \cdot \text{input}$$

$$MV = K_p \cdot e(t) \dots\dots\dots(2.8)$$

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



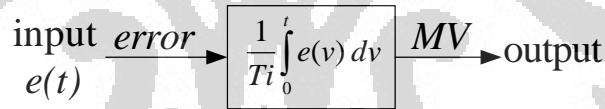
**Gambar 2.14. Grafik respon Pengendali *Proportional*[3]**

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. Dan jika nilai  $K_p$  kecil, sistem pengendali hanya mampu melakukan koreksi kesalahan yang kecil, sehingga akan menghasilkan respon sistem yang lambat.

**b. Pengendali *Integral* (I)**

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

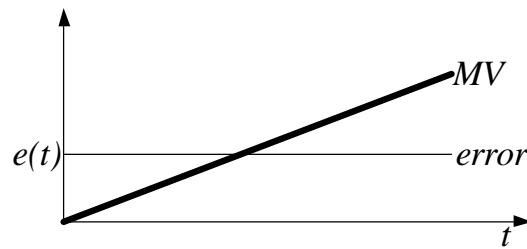


**Gambar 2.15. Blok Diagram Pengendali *Integral*[3]**

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

$$MV = \frac{1}{T_i} \int_0^t e(v) dv \dots\dots\dots(2.9)$$

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.



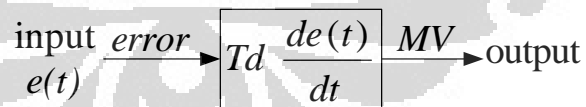
**Gambar 2.16. Grafik Respon Pengendali *Integral*[3]**

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*.

**c. Pengendali *Diferensial* (D)**

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



**Gambar 2.17. Blok Diagram Pengendali *Diferensial*[3]**

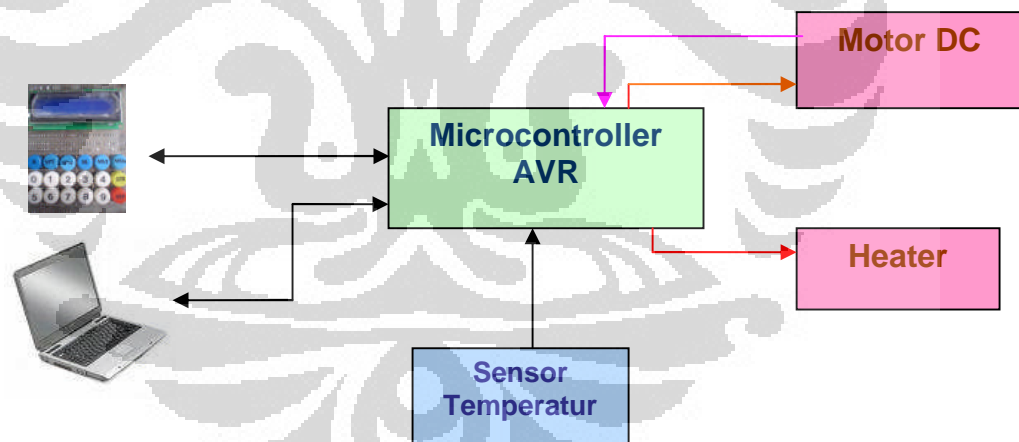
## BAB 3

### PERANCANGAN DAN CARA KERJA SISTEM

Pada bab ini akan dibahas mengenai perancangan sistem beserta cara kerja dari masing-masing *hardware* serta *software* yang digunakan penulis dalam penyusunan alat ‘Sistem Pengendali Temperatur dan Kecepatan *Shaking Water Bath* Berbasis *Microcontroller*’.

#### 3.1. Perancangan Kerja Alat

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

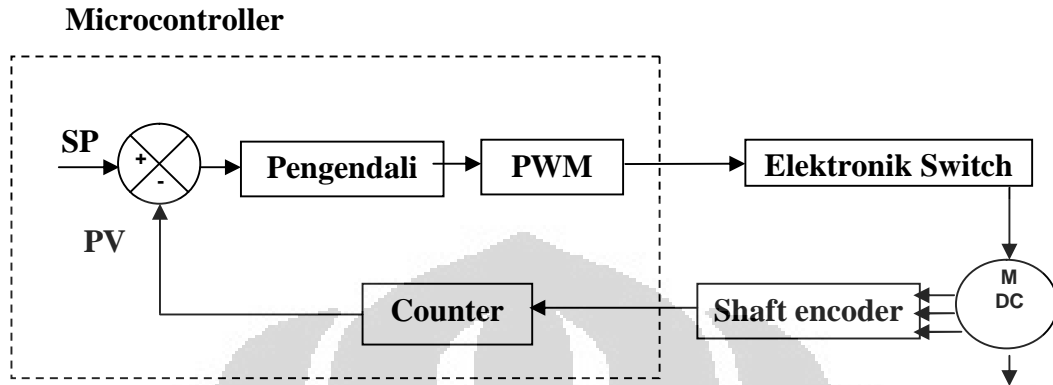


**Gambar 3.1. Blok diagram cara kerja alat**

Dari blok diagram di atas terdapat PC atau keypad sebagai data *transceiver* untuk mengirimkan nilai *Set Point* (SP) yang berupa bilangan ke dalam *microcontroller* dan menampilkan nilai *Process Variable* (PV). Proses pengendalian motor, *heater* dan temperaturnya terjadi di dalam *microcontroller*.

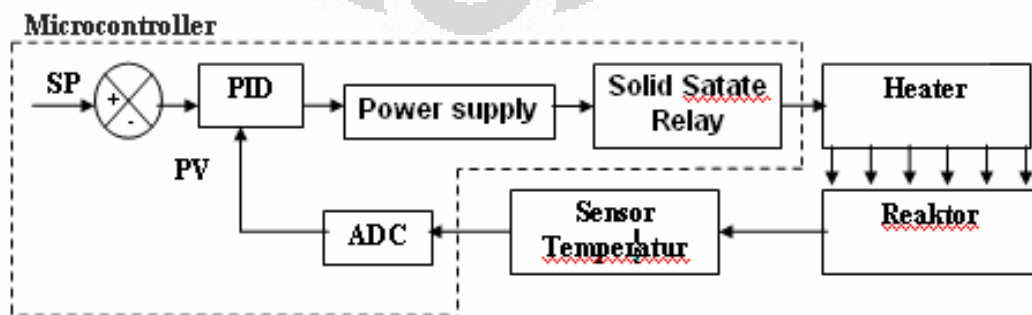


Pada dasarnya, alat ini dikendalikan oleh dua buah sistem pengendali yaitu pengendali kecepatan dan pengendali temperatur.



**Gambar 3. 2. Blok Diagram Pengendali Kecepatan Motor DC**

Setpoint (SP) dimasukkan dari rangkaian keypad. Rangkaian tersebut merupakan miniatur sistem yang menggunakan AT89S52. Setpoint dimasukkan melalui keypad dan kemudian di kirim melalui komunikasi serial untuk di proses di dalam microcontroller AVR yang berfungsi sebagai pengendali. Setpoint tersebut di kendalikan, dan output pengendaliannya masuk ke PWM (*Pulse Width Modulation*). Selanjutnya output PWM masuk ke dalam electronic switch untuk menggerakkan motor. Dari putaran motor, pulsa-pulsa dibaca oleh *shaft encoder* dan kemudian menjadi input counter. Counter tersebut nantinya akan kembali ke pengendali dalam bentuk *Manipulated Variable* (MV). Untuk bagian pengendali, PWM dan *counter* terdapat di dalam *microcontroller*.



**Gambar 3. 3. Blok Diagram Pengendali Temperatur**

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

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

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

$$\text{Output}_{\text{PID}} = K_p[E + K_i \sum E \Delta t + K_D \frac{\Delta E}{\Delta t}] \dots\dots\dots(3.2)$$

Nilai output PID tersebut akan mengatur keluaran *power supply* untuk mengatur sistem pemanasnya yaitu *heater* dengan mengendalikan *Solid State Relay* yang berfungsi sebagai saklar. Dengan demikian reaktor mengalami pemanasan.

Di dalam reaktor tersebut terdapat sensor temperatur, karena temperatur yang diharapkan dapat mengukur temperatur mencapai 100°C, maka menggunakan sensor temperatur termokopel tipe K, karena sensor temperatur ini untuk rentang suhu -200°C hingga 1200°C dan selain itu termokoel 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.

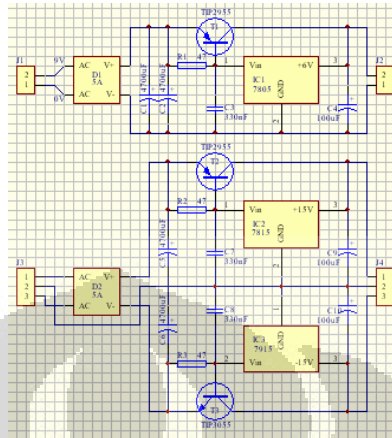
### 3.2 Perancangan Perangkat Keras (Hardware)

Di dalam pembuatan alat pengendali kecepatan putaran pengaduk dan temperatur pada pencampur otomatis ini perangkat kerasnya (*hardware*) terdapat beberapa bagian, yaitu:

#### 3.2.1. Rangkaian Power Supply

Rangkaian *power supply* digunakan untuk mengubah tegangan AC 220 volt dari PLN menjadi sebesar 24 volt, 5 volt, +15 volt dan -15 volt tegangan DC yang dibutuhkan untuk input rangkaian *minimum system microcontroller*,

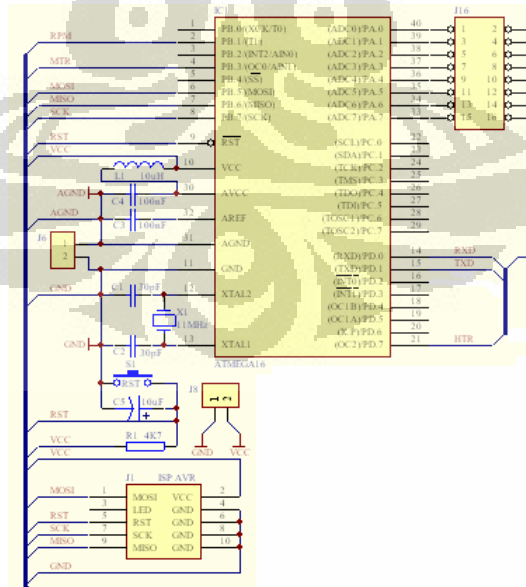
rangkain motor DC, *heater* dan rangkaian sensor temperatur. Di dalam *Power Supply* ini menggunakan trafo dan regulator.



**Gambar 3.3 Rangkaian Power Supply**

### 3.2.2. Rangkaian Minimum System

Alat ini menggunakan *microcontroller* ATmega16 untuk pengendaliannya. Di dalam rangkaian *minimum system* menggunakan kristal 11 MHz dan ada ISP AVR

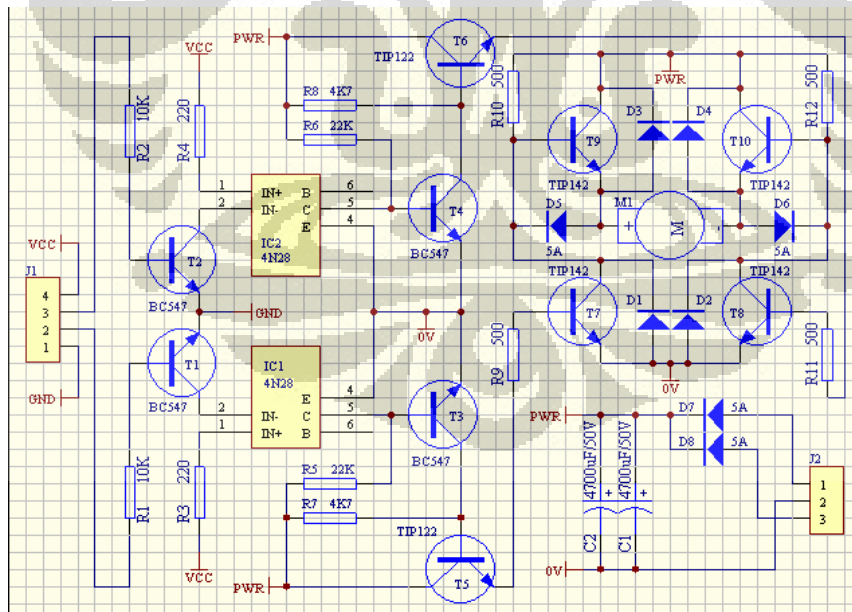


**Gambar 3.4 Minimum System Microcontroller AVR**

Rangkaian ini dibutuhkan agar dapat mengisi (*me-download*) program yang dibuat. Program tersebut meliputi program pengendalian temperatur dan kecepatan serta pengiriman data komunikasi serial. Port yang digunakan pada *microcontroller* ATmega16 adalah PB untuk RPM dan Motor lalu PD untuk sensor temperatur dan *heater*.

### 3.2.3. Rangkaian Pengendali Motor DC

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 di dalam motor DC juga terdapat sensor. Transistor pada rangkaian pengendali motor dc ini digunakan sebagai saklar elektronik (*electronic switch*). Pada gambar 3.6 merupakan rangkaian pengendali yang dapat mengendalikan kecepatan putaran sebuah motor dc. Selain itu juga dapat mengatur arah putar motor.

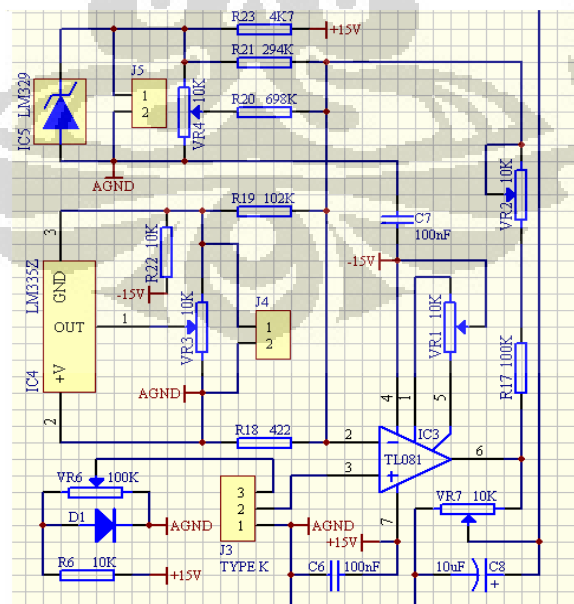


Gambar 3.5 Rangkaian *Driver* Motor DC

Tegangan yang diberikan sebagai *supply* motor yaitu sebesar 24 volt. Input dari pengendali motor dc ini merupakan sinyal yang dihasilkan dari pengendali *PWM* dan pengendali *count up compare*.

### 3.2.4. Rangkaian *Cold Junction*

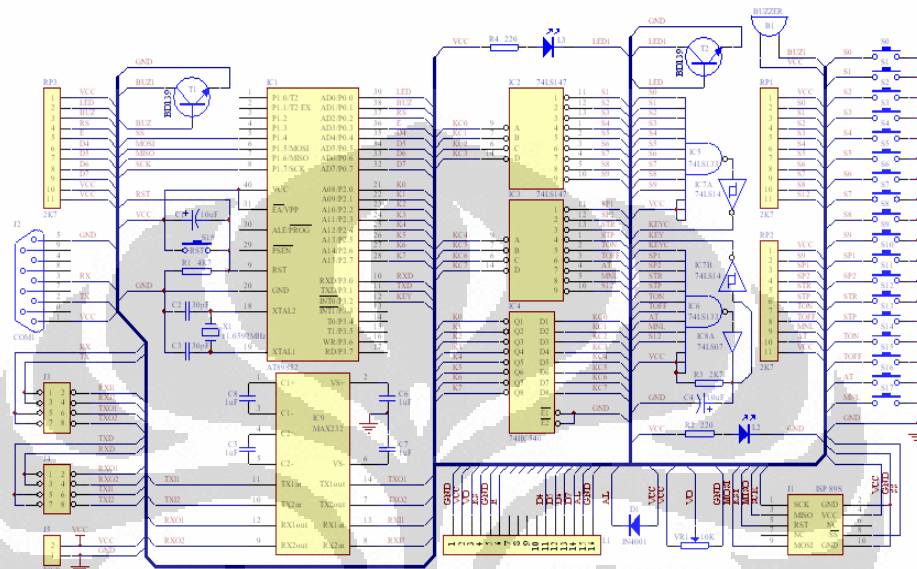
Termokopel yang sebelum digunakan perlu dikalibrasikan terlebih dahulu. Oleh karena itu menggunakan rangkaian *Cold Junction* untuk mengkalibrasinya. IC LM329DZ, LM335, dan termokopel (AGND dan input positif untuk TL081) harus dihubungkan singkat, lalu ukur tegangan output yang disebut tegangan *offset*. Tegangan offset diatur menjadi 0 V. Setelah itu hubungan singkat termokopel di atur kembali (kaki negatif termokopel tidak diberi AGND), untuk mendapatkan input sebesar 10mV dan pada output diatur hingga tegangan sebesar 2,48V. Hubung singkat lagi pada termokopel dan LM329, pada LM335 di atur hingga tegangan outputnya sebesar 3 V pada waktu temperatur ruang, yaitu 27 °C. Selanjutnya hubung singkat LM329DZ di buka dan di atur kembali hingga tegangan outputnya sebesar 0,27 V. Dengan demikian di dapat perbandingan antara temperatur dan tegangan. Dimana besar temperatur akan sama dengan besar tegangan, dengan kenaikan 10mV/°C.



**Gambar 3.6 Rangkaian *Cold Junction***

### 3.2.5 Rangkaian Keypad

Rangkaian Miniatur Sistem dan Keypad dapat dilihat pada gambar berikut ini :



**Gambar 3. 7 Rangkaian Keypad**

Rangkaian minsis ini menggunakan IC mikrokontroller AT89S52, alasan penulis memakai AT89S52 karena memori pada IC ini lebih besar disamping itu juga IC ini mempunyai tiga buah timer. Ketiga Timer penulis gunakan untuk melengkapi program pengendalian pada *Shaking Water Bath*, yang berfungsi sebagai timer PWM, timer pewatu sampling, dan timer counter. Pada minsys rangkaian ini terdiri dari rangkaian *keypad* yang digunakan untuk mengirim data kepada mikrokontroller Atmega16 atau rangkaian *minsys* AVR, pengiriman data yang digunakan berupa pengiriman melalui serial yang dilakukan oleh rangkaian serial MAX232, selain itu terdapat juga beberapa IC yang digunakan untuk proses *interrupts*.

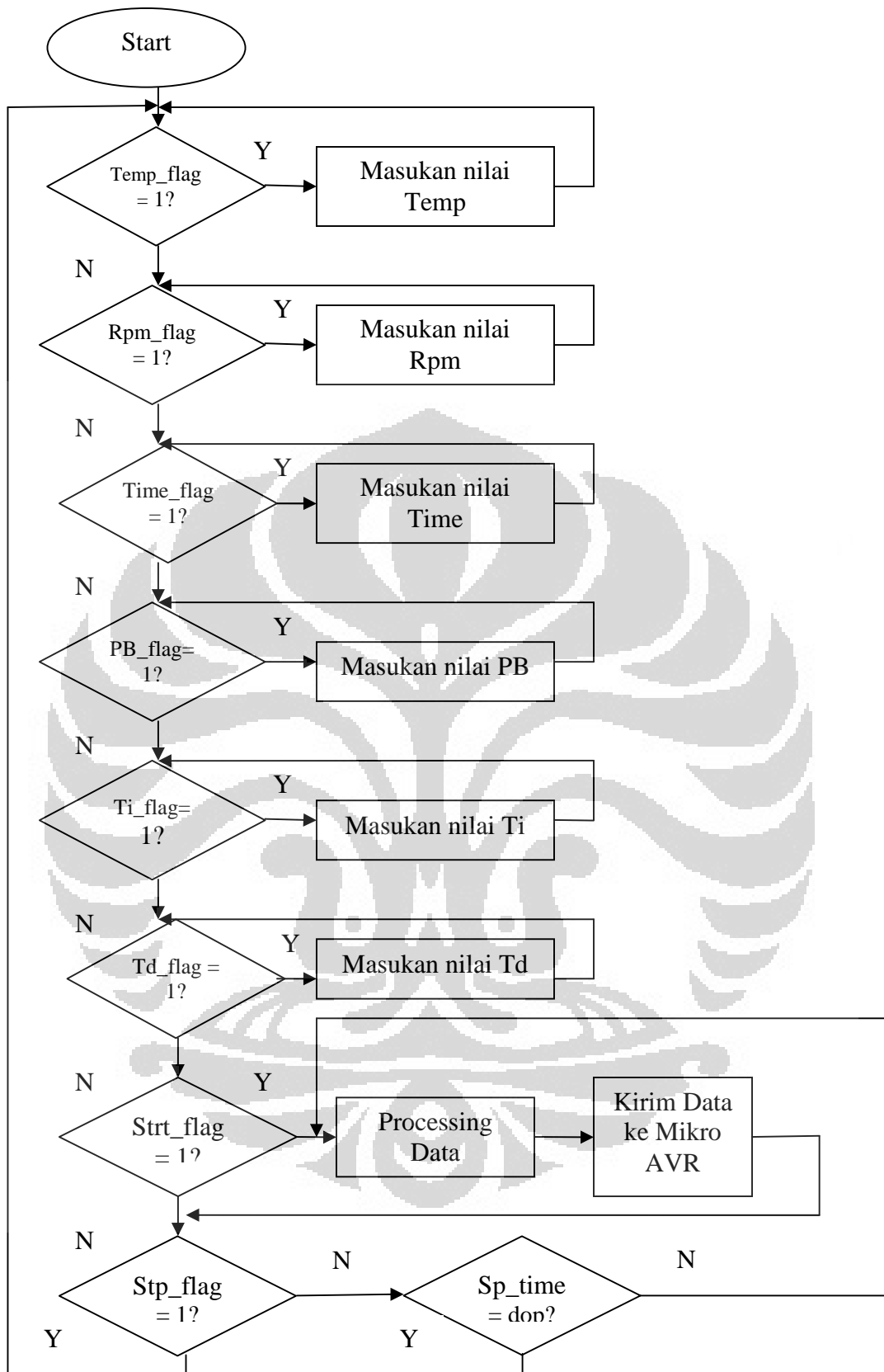
Rangkaian keypad ini terdapat 18 buah keypad yang mempunyai fungsi-fungsi berbeda yaitu sebagai fungsi angka 0 sampai 9 dan juga sebagai fungsi untuk memberikan perintah Temperatur, Rpm, Time, Pb, Ti, Td, STR dan STOP.

### 3.3 Perancangan Perangkat Lunak (Software)

Pada alat ini menggunakan perangkat lunak (*software*) di dalam pengendaliannya. *Software* yang digunakan yaitu Bascom AVR dan pengambilan data menggunakan LabVIEW. Setelah mendapatkan model dan mengetahui fungsi tranfers dari data temperatur dengan menggunakan metode *Kurva Reaksi* maka dibuat program Bascom AVR untuk mengendalikan sistem tersebut. Selain itu menggunakan kabel serial untuk komunikasinya, baik komunikasi dengan PC ataupun dengan rangkaian *keypad*.

#### 3.3.1 Flowchart Proses pada Rangkaian Keypad

Flowchart di bawah dapat dijelaskan bahwa pada awal proses, program akan menginisialisai variable-variabel yang digunakan pada program keypad tersebut, pada keadaan awal ini LCD akan menampilkan tulisan pada upperline "PID CONTROLLER", dan pada lowerline "V:1.2 FISIKA UI", lalu proses selanjutnya dengan memasukan nilai parameter Temp, Rpm, Time, PB, Td, Ti. Proses pemassukan nilai tersebut berjalan secara urut. Untuk PB, Td, dan Ti berfungsi agar proses respon output pada pengendalian *heater* ini memiliki respon yang akurat dan stabil, Setelah memasukan nilai parameter PID tersebut maka nilai-nilai parameter yang dimasukkan tersebut akan disimpan, selanjutnya menekan tombol. Ketika tombol "START" di tekan, maka seluruh proses kerja alat akan berjalan. Apabila alat sedang bekerja, perubahan masukan hanya dapat dilakukan untuk perubahan Temperatur, Rpm, dan Time. Untuk nilai PB, Ti dan Td hanya dapat dilakukan sebelum proses berjalan. Saat proses berjalan, data dikirimkan ke Bascom 89S52 untuk di tampilkan pada LCD. Sehingga dapat kita perubahan data saat proses berjalan dapat diketahui. Dan apabila tombol "STOP" ditekan maka keseluruhan proses akan berhenti. Motor dan *heater* dapat berhenti bekerja apabila waktu (DOP) habis dan atau tombol "STOP" di tekan. Apabila tombol "START" di tekan, .Shaking Water Bath dapat langsung di jalankan tanpa mengatur Temperatur, Rpm, dan time, karena pengaturan pengendali sebelumnya sudah tersimpan di *memory* dan dapat langsung di panggil.

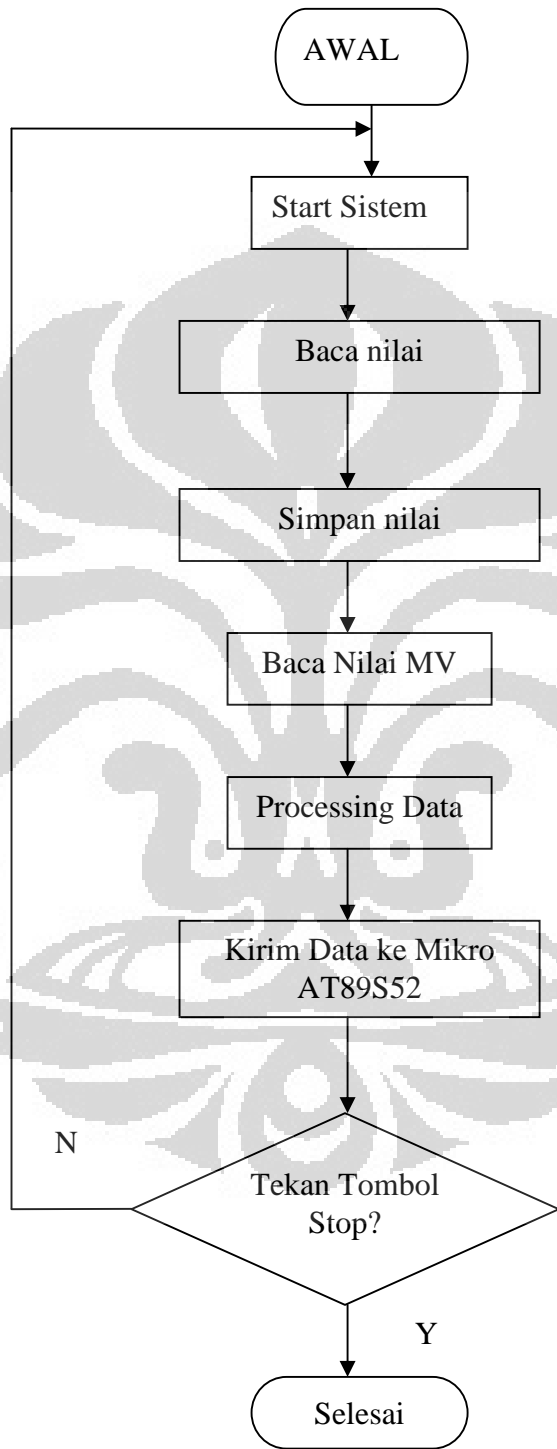


**Gambar 3.8. Flowchart program Keypad**



### 3.3.2 Flowchart Proses pada Rangkaian Minimum System ATmega16

Pada flowchart di bawah 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, RPM, Time, PB, Ti, Td yang sudah tersimpan pada *eprom*, maka data-data tersebut akan di olah untuk di proses. PB, Ti, dan Td yang telah di atur aka diolah AVR ke dalam rumus persamaan PID, sedangkan data-data temperatur, Rpm, dan time akan langsung dibandingkan untuk dijalankan karena didalam program AVR telah terdapat persamaan yang telah diambil dari pengambilan data alat sebelumnya. Data-data pada *eprom* adalah data inputan dari *keypad*. Apabila sudah membaca data pada *eprom* 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 di lakukan 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 *Microcontroller*. Akan tetapi untuk motor, pengendalian dilakukan dengan mengatur pulsa yang terdapat pada motor DC yang menggunakan PWM.



### Gambar 3.9 *Flowchart* program Pengendali

## BAB 4

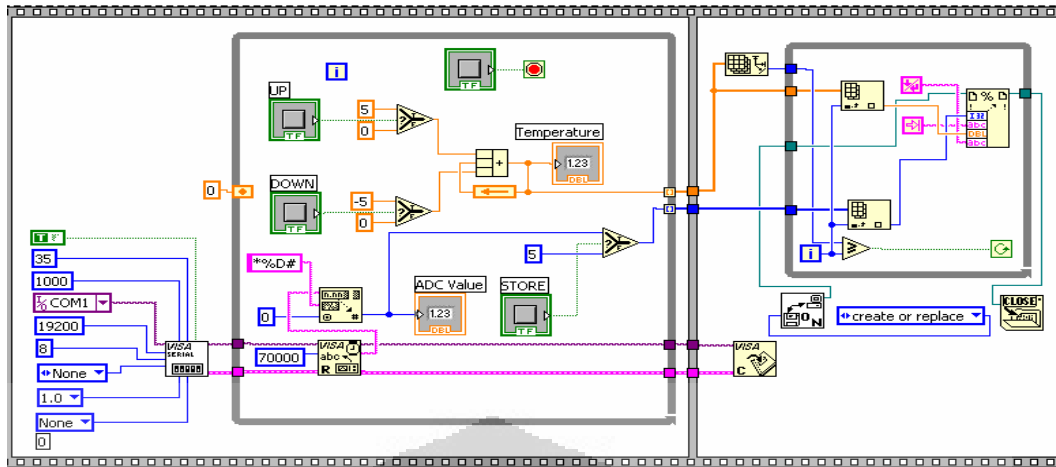
### DATA PERCOBAAN DAN ANALISA

Pembahasan pada bab 4 laporan tugas akhir penelitian ini adalah mengenai pengujian sistem dan analisa data penelitian. Dalam bab ini, akan dibahas tentang bagaimana proses dan cara pengambilan data penelitian yang dilakukan penulis. Pengujian-pengujian tersebut meliputi :

- Pengujian Termokopel
- Pengujian Motor
- Pengujian Sistem Pengendalian Dengan Kurva Reaksi

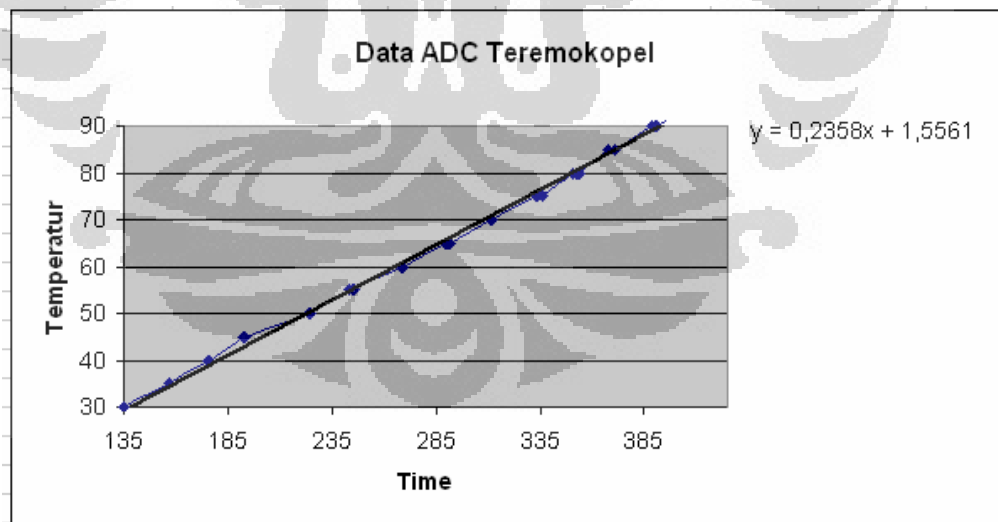
#### 4.1. Pengujian Termokopel

Pengujian Termokopel bertujuan untuk mengetahui besar ADC yang diukur oleh termokopel bila pada *heater* diberikan daya mencapai 100%. Pengujian dilakukan menggunakan air untuk mendapatkan *range* temperatur hingga 100°C. Pengujian dilakukan sebanyak 5 kali percobaan dari 30 °C hingga 100 °C setiap perubahan 5 °C dengan naik dan turun. Pengukuran temperaturnya diukur dan dibandingkan oleh termometer alkohol dan temperatur yang pada termokopel menggunakan LabVIEW dalam bentuk nilai desimal dari bit ADC. Program LabVIEW untuk pengambilan data ADC, yaitu :



**Gambar 4.1 LabVIEW untuk pengambilan data ADC**

Dari LabVIEW di atas, dapat diketahui nilai ADC dari termokopel yang dibandingkan dengan temperatur yang terukur pada termokopel. Pengambilan data ADC dilakukan sebanyak 10 kali. Dan dari seluruh percobaan, didapat grafik dengan persamaan yang paling mendekati linier. Dimana nilai ADC sama dengan nilai temperatur yang terukur pada termometer. Grafik tersebut sebagai berikut :



**Gambar 4.7 Grafik Persamaan ADC**

Dari grafik terlihat bahwa perubahan temperaturnya hampir linear dengan persamaan garis, yaitu:

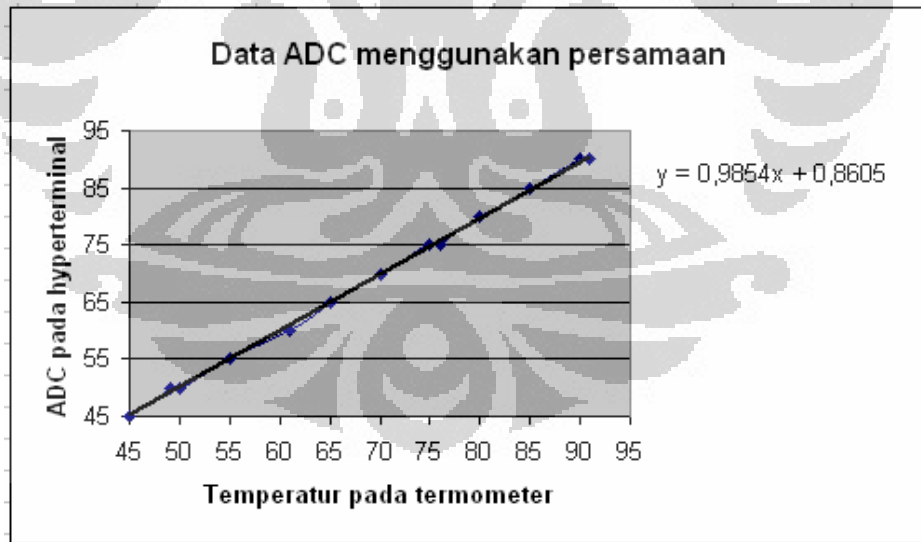
$$y = 0.2358x + 1.5561 \dots\dots\dots(4.1)$$

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

$$x = \frac{y - 1.5561}{0.2358} \dots\dots\dots(4.2)$$

Dimana x adalah nilai temperatur yang terukur pada termometer digital dan y adalah nilai bit dari ADC.

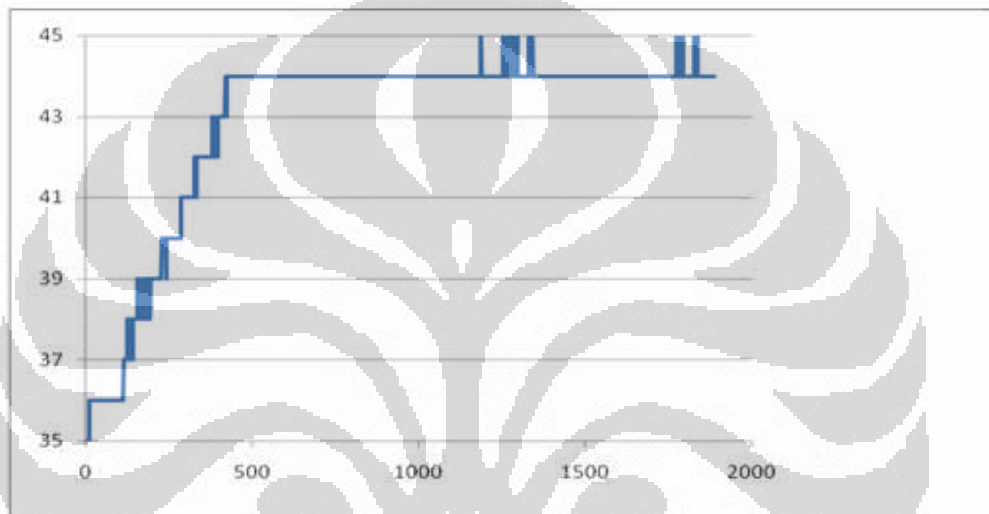
Setelah didapatkan rata-rata temperatur maka diuji kembali dengan mencoba membandingkan data antara ADC temperatur pada *hyperteminal* dengan temperatur yang terbaca pada termometer. Perbandingan ini dilakukan untuk membuktikan bahwa persamaan diatas adalah benar. Dengan demikian didapatkan grafik:



**Gambar 4.8 Grafik perbandingan antara nilai ADC *hyperteminal* pada komputer dengan temperatur pada termometer**

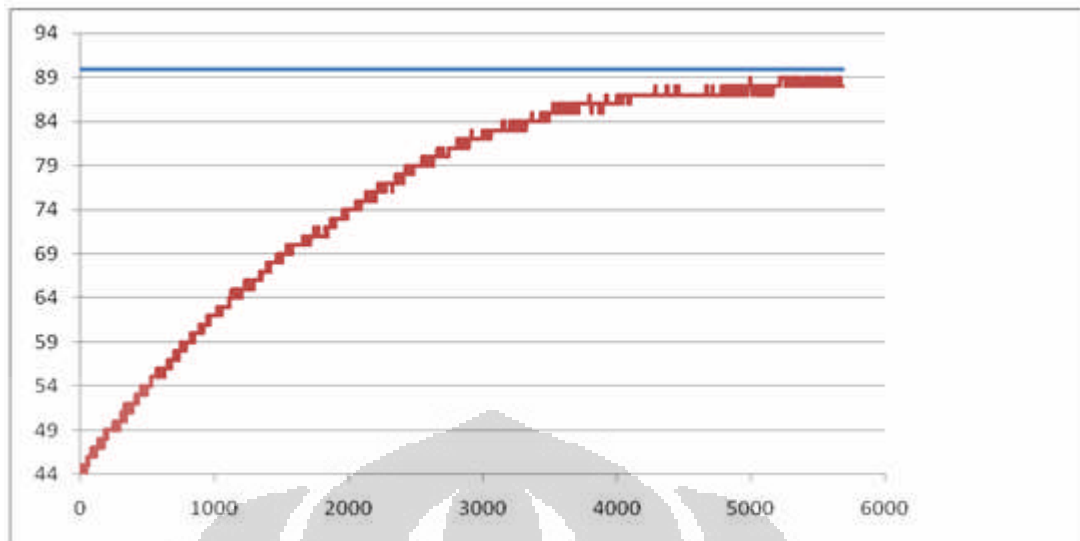
Dari Grafik di atas dapat diketahui bahwa persamaan ADC tersebut menghasilkan grafik yang mendekati linier. Sehingga nilai ADC pada hyperterminal pada komputer akan sebanding dengan nilai temperatur yang terukur pada termometer air raksa.

Setelah didapatkan persamaan temperatur tersebut, maka diuji kembali dengan pengujian daya. Yaitu dengan memanaskan heater dengan daya 10%. Dengan demikian didapatkan grafik:



**Gambar 4.9 Respon waktu proses perubahan temperatur dengan daya 10%**

Dari grafik di atas terlihat bahwa temperatur mengalami kestabilan saat mencapai temperatur 44 °C. Sehingga daya 10% hanya mampu mencapai temperatur maksimal 44 °C. Kemudian daya di naikan kembali sebesar 90%. Sehingga menghasilkan grafik, yaitu:



**Gambar 4.10 Respon waktu proses perubahan temperatur dengan daya 90%**

Dari grafik di atas terlihat bahwa temperatur 88 °C adalah titik stabil temperatur pada termokopel dengan daya 90%. Temperatur maksimal yang mampu dicapai, tidak sampai 90 °C. Hal tersebut dapat disebabkan oleh penguapan yang begitu besar saat proses pemanasan terjadi.

#### 4.1 Pengujian Motor

Pengujian motor DC bertujuan untuk mengetahui nilai respon yang dihasilkan oleh putaran motor dengan input tegangan yang berbeda. Tegangan motor 0 - 24 V. Dari *range* tegangan tersebut, dapat dilakukan pengaturan tegangan untuk mengatur kecepatan putar motor. Akan tetapi dapat juga dilakukan dengan pengaturan lebar pulsa pada motor yang terukur oleh *shaft encoder*.

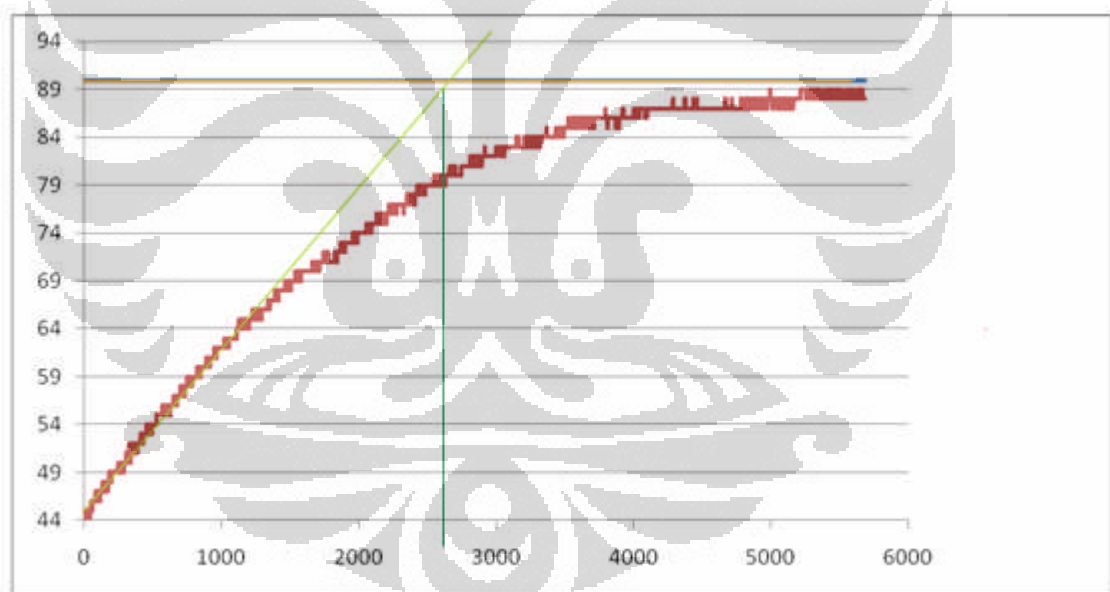
Pengaturan lebar pulsa dilakukan menggunakan PWM (*Pulse Width Modulation*), dimana cara ini memanfaatkan besar persentase *duty cycle* pada pulsa. Semakin banyak pulsa *high* pada motor, maka kecepatan motor akan semakin tinggi. Dan apabila pulsa *low* lebih banyak, maka motor akan cenderung lambat.

Banyaknya putaran pada motor dihitung selama satu menit, sehingga didapat banyaknya putaran setiap menit atau dengan kata lain adalah RPM (*Radian Per Menit*). Besar RPM yang didapat akan dibagi dengan *counter* yang

kemudian dimasukkan ke dalam Bacom AVR yang digunakan sebagai pengendali. *Setpoint* yang berasal dari Bacom 8052, akan di *compare* dengan sensor, sehingga kecepatan motor DC menjadi stabil.

### 4.3 Pengujian Sistem Pengendalian Dengan Kurva Reaksi

Pengujian sistem pengendali dengan metode kurva reaksi berfungsi mencari fungsi transfer dengan mencoba sistem tersebut, bukan dengan cara *trial* dan *error* (pengamatan dan penujian sistem pengendali PID yang dilakukan secara coba-coba dengan memasukkan nilai-nilai yang pada sistem sampai sistem menjadi stabil). Dari Grafik daya 90% pada pengujian termokopel merupakan langkah yang dilakukan untuk menentukan nilai matematik dari respon sistem, untuk mendapatkan fungsi transfernya menggunakan metode kurva reaksi. Kurva reaksi yang dihasilkan adalah pada gambar berikut ini:



**Gambar 4.11 Kurva Reaksi Dari Respon Sistem**

Bila dilihat grafik di atas maka dapat diketahui fungsi transfer dari sistem dengan menggunakan *reaction-curve method* (metode kurva reaksi), didapat *Lag time* dan *time*, dimana  $Lag = L = 62.5$  dan  $time = T = 2438$ . sehingga slope kurva dapat diketahui dengan :



$$N = \frac{\Delta PV}{T} \dots\dots\dots(4.3)$$

Dimana  $\Delta PV = 88 - 44 = 44$  , maka nilai  $N = 0.018$ . Dengan demikian dapat diketahui  $K_p$ , yaitu :

$$K_p = \frac{1.2\Delta CV}{NL} \dots\dots\dots(4.4)$$

Dari grafik,  $\Delta CV = 90 - 10 = 80$ , maka nilai  $K_p = 85.3$ . Selain  $K_p$ , nilai  $K_i$  dan  $K_d$  juga dibutuhkan untuk membangun persamaan PID. Sehingga diketahui bahwa :

$$K_i = \frac{1}{2L} \dots\dots\dots(4.5)$$

$$K_d = 0.5 L \dots\dots\dots(4.6)$$

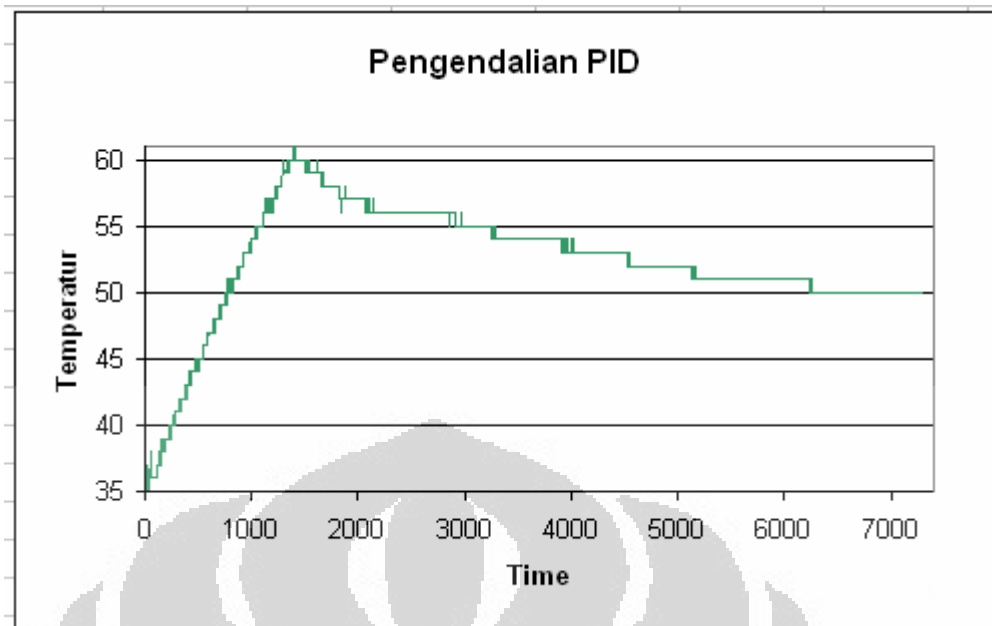
Nilai dari persamaan di atas yaitu  $K_i = 0.008$  dan  $K_d = 31.25$ . Dari  $K_i$  juga kita dapat menentukan sigma error dengan rumus :

$$SE = \frac{100}{K_i} \dots\dots\dots(4.7)$$

*Sigma error* dari percobaan ini adalah  $SE = 12500$ . Dari semua nilai yang telah diketahui, yaitu  $K_p$ ,  $K_i$ ,  $K_d$  dan  $SE$ , selanjutnya memasukkan nilai-nilai tersebut ke dalam program pengendalian pada Bascom AVR. Dari nilai tersebut juga dapat diketahui nilai  $PB$ ,  $T_i$ , dan  $T_d$ . Sehingga pengendalian manual dengan memasukkan nilai  $PB$ ,  $T_i$ , dan  $T_d$  juga dapat dilakukan.

Dari nilai-nilai yang telah diketahui tersebut, dilakukan pengujian sistem secara keseluruhan untuk melihat respon sistem menggunakan pengendalian PID.

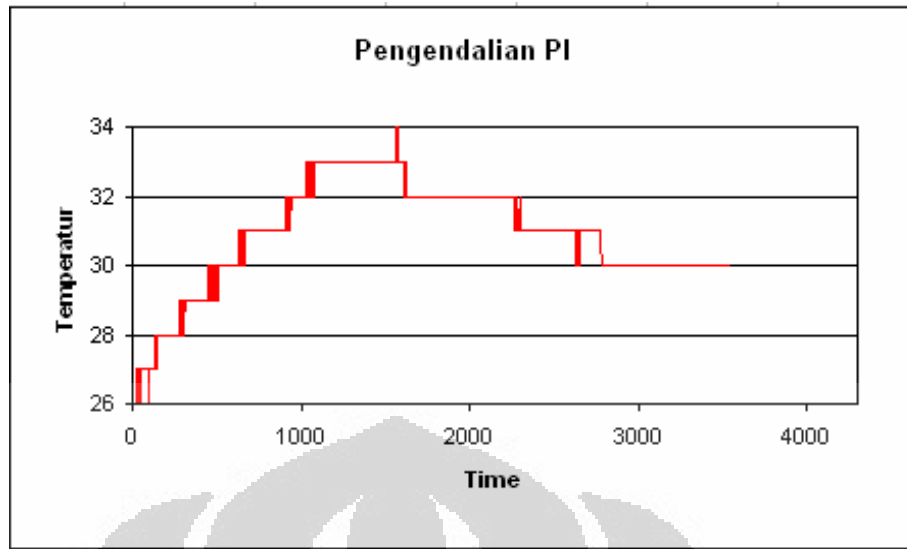
Untuk setpoint temperatur =  $50^\circ C$ , sistem mengalami *overshoot* sebesar  $10^\circ C$ . Setelah itu, temperatur turun dan kemudian mencapai kestabilan dengan time yang cukup besar akibat *overshoot* yang sangat tinggi.



**Gambar 4.12 Grafik persamaan PID dengan setpoin 50 °C**

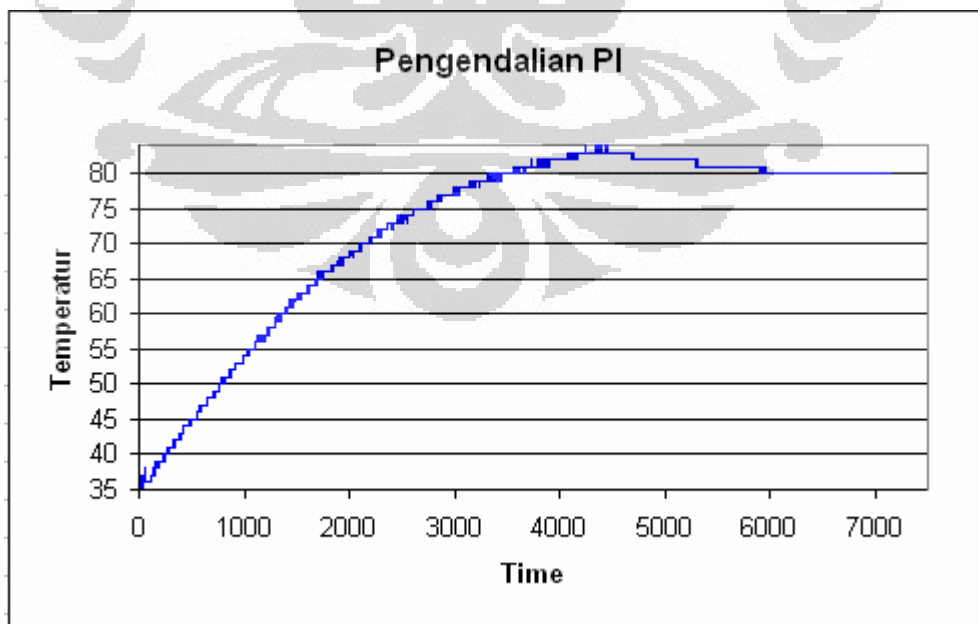
Dari *overshoot* yang begitu besar, maka dapat diperkirakan bahwa nilai  $K_d$  yang dimasukkan begitu besar, yaitu mencapai 31.25. Sehingga dalam persamaan nilai  $K_d$  dijadikan 0 untuk mengurangi *overshoot*.

Selanjutnya untuk pengujian sistem keseluruhan dengan nilai  $K_d = 0$  pada setpoin = 30 °C, sistem mengalami *overshoot* mencapai 3 °C. Nilai tersebut jauh lebih kecil dibandingkan dengan *overshoot* dengan nilai  $K_d$  yang besar. Akan tetapi *lag time* yang dihasilkan lebih besar. Sehingga sistem berjalan relatif lebih lambat untuk mencapai setpoin. Tetapi untuk mendapat kestabilan, *time* yang dibutuhkan tidak besar. Dan kestabilan dapat dicapai dengan relatif cepat. Dengan kata lain, mengubah persamaan menjadi bentuk PI membuat sistem berjalan dengan *overshoot* yang kecil



**Gambar 4.13 Grafik persamaan PI dengan setpoint 30 °C**

Begitu pula untuk setpoint = 80 °C, sistem mengalami *overshoot* mencapai 3 °C. Setelah mencapai *overshoot* tersebut, temperatur kembali turun dan menjadi stabil. Lama waktu untuk mencapai kestabilan cukup besar. Akan tetapi setelah mencapai setpoint, temperatur akan stabil. Hal tersebut disebabkan oleh pengendalian yang diatur melalui *Solid State Relay*.



#### Gambar 4.11 Grafik persamaan PI dengan setpoint 80 °C

Kedua grafik di atas membuktikan bahwa sistem berjalan dengan baik. Karena dihasilkan temperatur yang stabil. Dan metode kurva reaksi membuat sistem bekerja sesuai dengan sistemnya karena perhitungan dapat dilakukan secara tepat, disesuaikan dengan nilai matematis sistem. Akan tetapi penggunaan PID membuat *overshoot* semakin besar dibandingkan dengan PI.



## BAB 5

### KESIMPULAN DAN SARAN

Bab 5 merupakan bab terakhir yang ditulis oleh penulis pada penyusunan dan penulisan laporan tugas akhir penelitian yang membahas tentang kesimpulan dari penelitian yang telah dilakukan dan saran yang ditulis, merupakan fakta dan kenyataan yang harus dihadapi penulis dalam proses pengendalian, pengujian sistem, dan pengambilan data pada saat berlangsungnya kegiatan penelitian.

#### **Kesimpulan**

Setelah menyelesaikan perancangan sistem serta pengujian terhadap sistem tersebut, maka penulis dapat mengambil suatu kesimpulan bahwa :

1. Alat ini berjalan dengan baik untuk mempercepat reaksi pencampuran dan pemanasan dengan mengatur temperatur dan kecepatan gerakanya.
2. Termokopel yang digunakan adalah termokopel tipe K diselubungi oleh bahan *stainless*.
3. Termokopel diperkuat dengan menggunakan rangkaian *cold junction*.
4. ADC yang digunakan 10 bit dengan tegangan referensi 2,56 volt.
5. *Time Sampling* pengendalian yang digunakan sebesar 1 s.
6. Dengan metode *reactive curve method* didapat persamaan pengendali berupa pengendali PID dengan nilai  $K_p = 85,3$  ,  $K_i = 0.008$  , dan  $K_d = 31.25$  , menghasilkan overshoot mencapai  $10\text{ }^\circ\text{C}$ .
7. Respon pengendalian yang dihasilkan dengan *reactive curve method* baik, dengan overshoot mencapai  $3\text{ }^\circ\text{C}$  , dengan memperkecil atau menghilangkan nilai  $K_d$  yang sebelumnya begitu besar.
8. Pengendalian menggunakan PI menghasilkan overshoot yang jauh lebih kecil dari pengendalian PID
9. Semakin banyak pulsa *high* pada motor, maka kecepatan motor akan semakin tinggi

10. Semakin besar tegangan input yang diberikan pada motor, maka banyak putaran motor akan semakin besar

 **Saran**

1. Sebaiknya membuat konstruksi yang sehingga tidak terjadi penguapan berlebih saat proses pemanasan terjadi, agar temperatur yang dicapai dapat mendekati 100 °C. Sehingga proses pemanasan dapat berjalan baik
2. Sebaiknya pengambilan data dilakukan dengan besaran yang tetap (tidak berubah) untuk beberapa kali pengambilan agar hasil perhitungan lebih presisi dengan fungsi tranfers yang tepat, sehingga dapat mengurangi *overshoot* yang cukup besar.



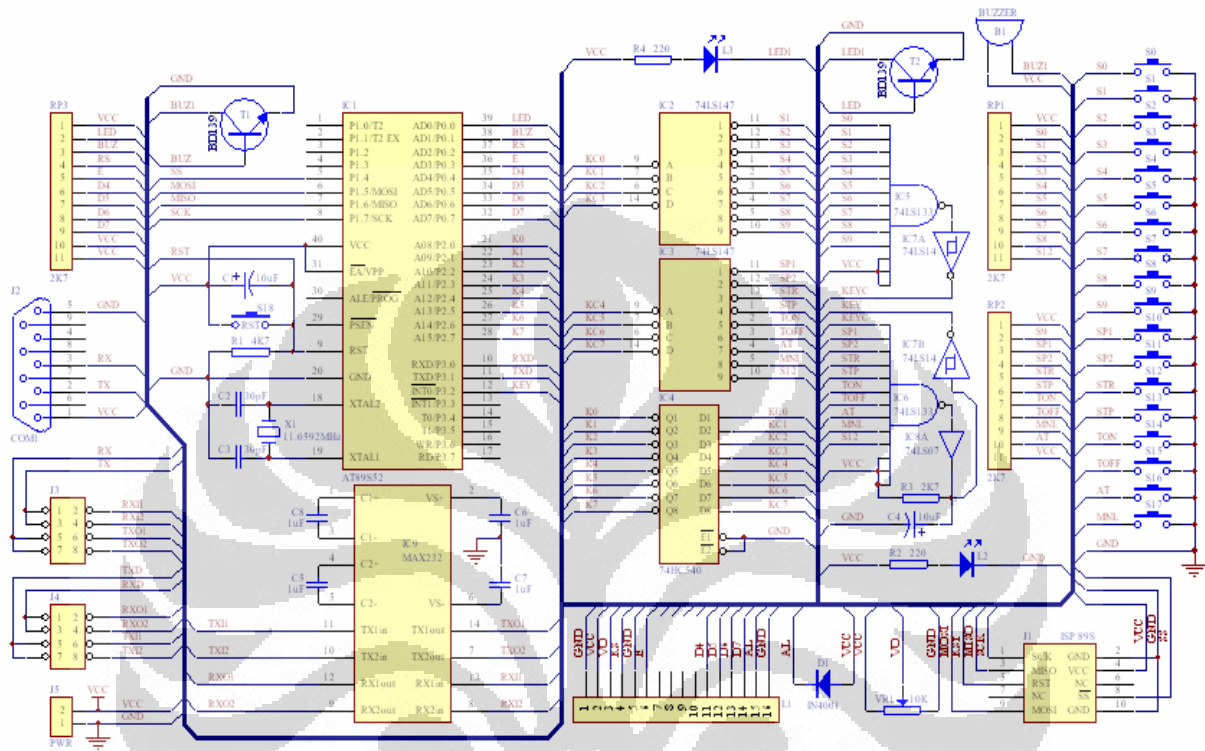
## DAFTAR PUSTAKA

1. Kilian. *Modern Control Technology: Components and Systems*. Delmar, Inc.
2. Putra, Agfianto Eko., “Belajar Mikrokontroler”, Penerbit Gava Media, 2003.
3. Kurniasih, Dewi. Laporan Tugas Akhir dengan Judul “Pengendalian Level Cairan Menggunakan Metode PID Berbasis Personal Computer (PC)”, Departemen Fisika, Universitas Indonesia, 2007
4. George Clayton, ”Operational Amplifiers”, Erlangga 2002.
5. Yamidi, Muhammad. Laporan Tugas Akhir dengan Judul “Rancang Bangun Sistem Pemanas Terkendali”, Departemen Fisika, Universitas Indonesia, 2007.



**LAMPIRAN A**  
**(RANGKAIAN-RANGKAIAN)**





## Rangkaian Keypad



**LAMPIRAN B**  
**(PROGRAM KONTROLLER DAN KEYPAD PADA**  
**BASCOM MICROCONTROLLER)**

## PROGRAM PENGENDALI

\$regfile = "m16def.dat"

\$crystal = 11059200

\$baud = 19200

Config Timer2 = Timer , Prescale = 8

Config Timer1 = Pwm , Pwm = 10 , Compare A Pwm = Clear Down , Compare B

Pwm = Clear Down , Prescale = 64

Config Timer0 = Counter , Edge = Falling

Config Adc = Single , Prescaler = Auto , Reference = Internal

Config Portd.6 = Output

Config Portd.7 = Input

Heater Alias Portd.6

Water\_level Alias Pind.7

Dim Error As Single

Dim Error\_b As Byte

Dim Sigma\_error As Single

Dim M\_variabel As Single

Dim Mvd As Single

Dim Error\_awal As Single

Dim D\_error As Single

Dim Pv\_rpm As Integer

Dim Konversi\_flag As Bit

Dim Balik As Bit

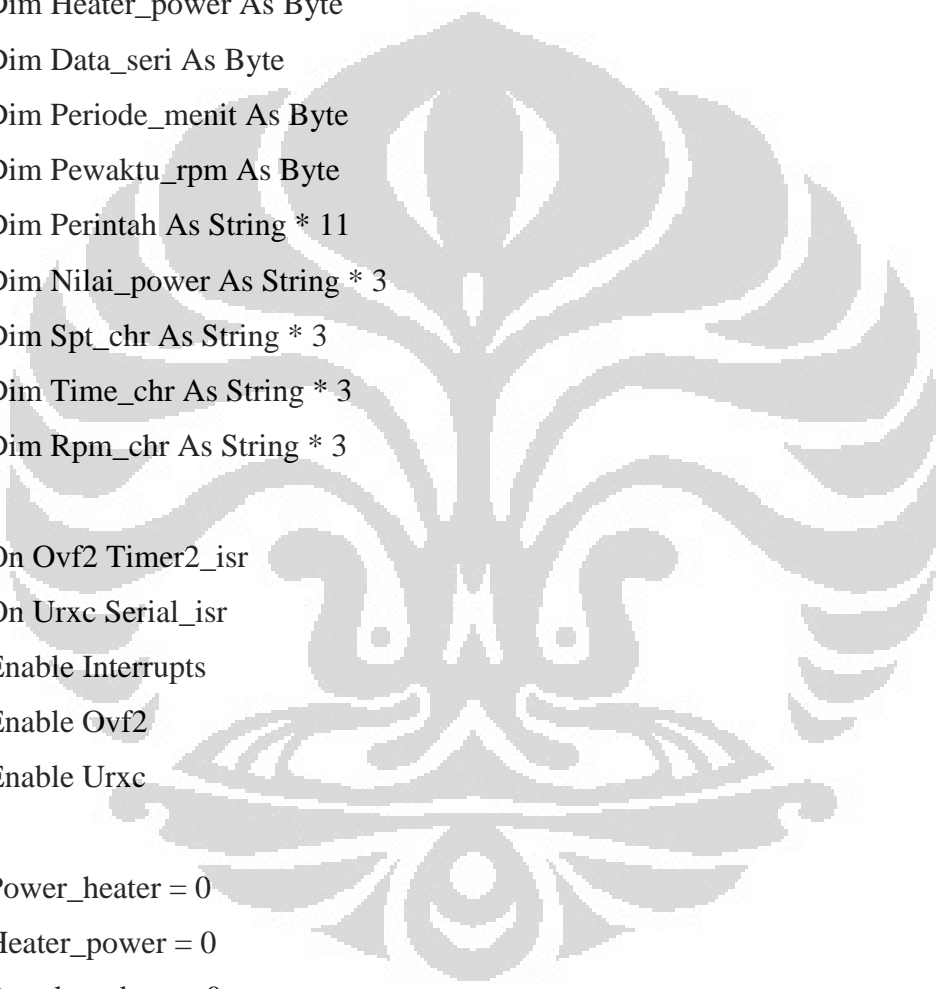
Dim Start\_prog\_flag As Bit

Dim Dop\_flag As Bit

Dim Kendali\_motor\_flag As Bit

Dim Spt As Word

Dim Temperature As Word



Dim Adc\_value As Word  
Dim Temp\_adc As Single  
Dim Pewaktu\_sampling As Word  
Dim Pewaktu\_daya As Word  
Dim Dop As Word  
Dim Rpm As Word  
Dim Power\_heater As Byte  
Dim Heater\_power As Byte  
Dim Data\_seri As Byte  
Dim Periode\_menit As Byte  
Dim Pewaktu\_rpm As Byte  
Dim Perintah As String \* 11  
Dim Nilai\_power As String \* 3  
Dim Spt\_chr As String \* 3  
Dim Time\_chr As String \* 3  
Dim Rpm\_chr As String \* 3

On Ovf2 Timer2\_isr  
On Urxc Serial\_isr  
Enable Interrupts  
Enable Ovf2  
Enable Urxc

Power\_heater = 0  
Heater\_power = 0  
Pewaktu\_daya = 0  
Pewaktu\_sampling = 0  
Konversi\_flag = 0  
Start\_prog\_flag = 0  
Balik = 1

Main\_prog:

```

Start Adc
Start Timer0
Start Timer1
Start Timer2
Pwm1a = 0
Pwm1b = 0
Do
Loop Until Balik = 0
If Start_prog_flag = 1 Then
  Balik = 1
  Error_awal = 0
  Do
    If Kendali_motor_flag = 1 Then
      Kendali_motor_flag = 0
      Pv_rpm = Pv_rpm / 40
      Pv_rpm = Pv_rpm * 60
      If Pv_rpm < Rpm Then
        Pwm1a = Pwm1a + 1
        Pwm1b = 0
      Else
        Pwm1a = Pwm1a - 1
        Pwm1b = 0
      End If
    End If
  End Do
  If Konversi_flag = 1 Then
    Konversi_flag = 0
    Adc_value = Getadc(0)
    Temp_adc = Adc_value * 0.2358
    Temp_adc = Temp_adc + 1.5561
    Temperature = Temp_adc
    Error = Spt - Temperature
  End If
End If

```

```

Error_b = Error
Sigma_error = Sigma_error + Error
If Sigma_error > 12500 Then Sigma_error = 12500
If Sigma_error < 0 Then Sigma_error = 0
D_error = Error - Error_awal
Error_awal = Error
M_variabel = Sigma_error * 0.008
M_variabel = M_variabel + Error
Mvd = 0 * D_error
M_variabel = M_variabel + Mvd
M_variabel = 85.3 * M_variabel
If M_variabel > 100 Then M_variabel = 100
If M_variabel < 0 Then M_variabel = 0
Heater_power = M_variabel
If Dop_flag = 1 Then
    Periode_menit = Periode_menit + 1
    If Periode_menit = 60 Then
        Periode_menit = 0
        Dop = Dop - 1
        If Dop = 0 Then
            Start_prog_flag = 0
            Heater_power = 0
            Balik = 0
        End If
    End If
End If
Print "*" ; Temperature ; ":" ; Pv_rpm ; ":" ; Dop ; "#"
End If
Loop Until Balik = 0
End If
Goto Main_prog

```

```

Timer2_isr:
Pewaktu_daya = Pewaktu_daya + 1
If Pewaktu_daya = 54 Then
    Pewaktu_daya = 0
    If Power_heater > 0 Then
        Power_heater = Power_heater - 1
        Set Heater
    Else
        Reset Heater
    End If
    Pewaktu_sampling = Pewaktu_sampling + 1
    Pewaktu_rpm = Pewaktu_rpm + 1
    If Pewaktu_rpm = 4 Then
        Pewaktu_rpm = 0
        Pv_rpm = Counter0
        Timer0 = 0
        Start Timer0
        Kendali_motor_flag = 1
    End If
    If Pewaktu_sampling = 100 Then
        Konversi_flag = 1
        Pewaktu_sampling = 0
        Power_heater = Heater_power
    End If
End If
Return

```

```

Serial_isr:
Disable Interrupts
Data_seri = Inkey()
If Data_seri = 42 Then
    Perintah = ""

```

```

Balik = 1
Do
  Data_seri = Waitkey()
  Print Chr(data_seri);
  If Data_seri = 58 Then
    Balik = 0
  Else
    Perintah = Perintah + Chr(data_seri)
  End If
Loop Until Balik = 0
If Perintah = "START" Then
  Balik = 1
  Spt_chr = ""
  Do
    Data_seri = Waitkey()
    Print Chr(data_seri);
    If Data_seri = 58 Then
      Balik = 0
    Else
      Spt_chr = Spt_chr + Chr(data_seri)
    End If
  Loop Until Balik = 0
  Spt = Val(spt_chr)
  Balik = 1
  Rpm_chr = ""
  Do
    Data_seri = Waitkey()
    Print Chr(data_seri);
    If Data_seri = 58 Then
      Balik = 0
    Else
      Rpm_chr = Rpm_chr + Chr(data_seri)

```



```

    End If
Loop Until Balik = 0
Rpm = Val(rpm_chr)
Balik = 1
Time_chr = ""
Do
    Data_seri = Waitkey()
        Print Chr(data_seri);
    If Data_seri = 35 Then
        Balik = 0
    Else
        Time_chr = Time_chr + Chr(data_seri)
    End If
Loop Until Balik = 0
Dop = Val(time_chr)
If Dop = 0 Then
    Dop_flag = 0
Else
    Dop_flag = 1
End If
Start_prog_flag = 1
Balik = 0
End If
If Perintah = "STOP" Then
    Heater_power = 0
    Pwm1a = 0
    Pwm1b = 0
    Start_prog_flag = 0
    Balik = 0
End If
If Perintah = "TEMPERATURE" Then
    Balik = 1

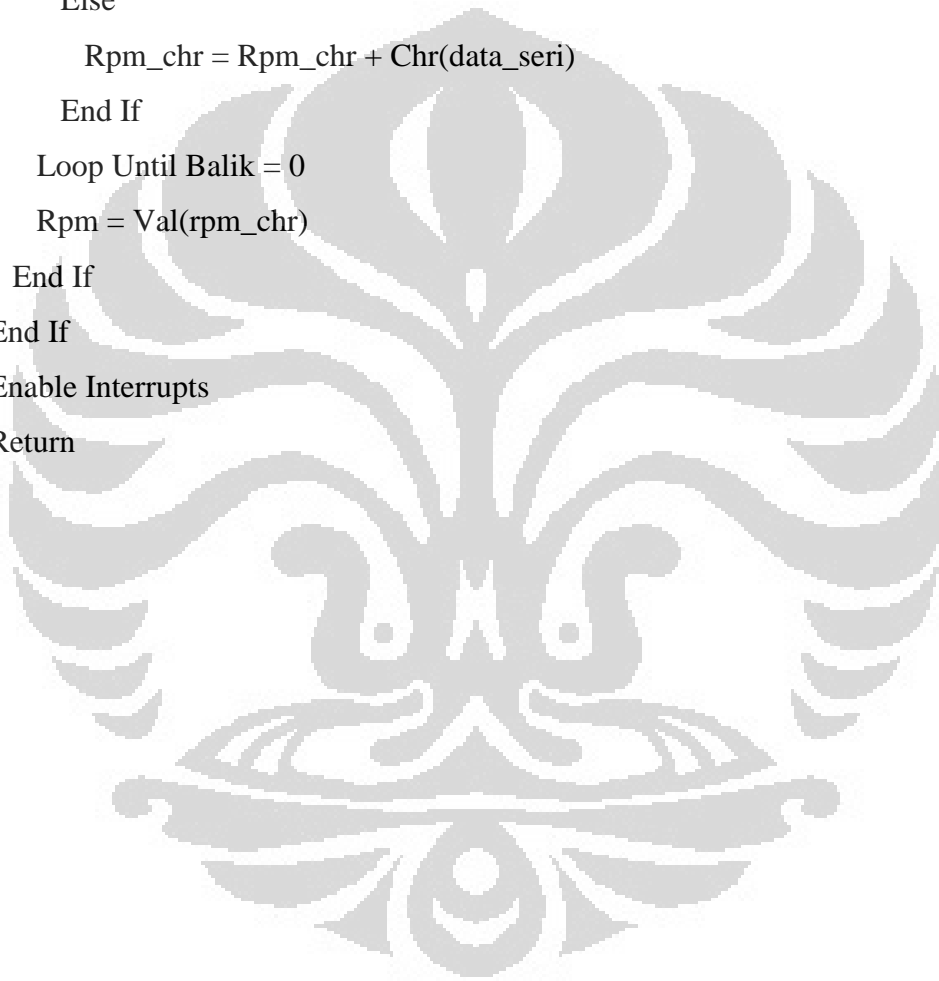
```

```

Spt_chr = ""
Do
  Data_seri = Waitkey()
  Print Chr(data_seri);
  If Data_seri = 35 Then
    Balik = 0
  Else
    Spt_chr = Spt_chr + Chr(data_seri)
  End If
Loop Until Balik = 0
Spt = Val(spt_chr)
End If
If Perintah = "TIME" Then
  Balik = 1
  Time_chr = ""
  Do
    Data_seri = Waitkey()
    Print Chr(data_seri);
    If Data_seri = 35 Then
      Balik = 0
    Else
      Time_chr = Time_chr + Chr(data_seri)
    End If
  Loop Until Balik = 0
  Dop = Val(time_chr)
  If Dop = 0 Then
    Dop_flag = 0
  Else
    Dop_flag = 1
  End If
End If
If Perintah = "RPM" Then

```

```
Balik = 1
Rpm_chr = ""
Do
  Data_seri = Waitkey()
  Print Chr(data_seri);
  If Data_seri = 35 Then
    Balik = 0
  Else
    Rpm_chr = Rpm_chr + Chr(data_seri)
  End If
Loop Until Balik = 0
Rpm = Val(rpm_chr)
End If
End If
Enable Interrupts
Return
```



## PROGRAM KEYPAD

\$regfile = "8052.DAT"

\$large

\$crystal = 11059200

\$baud = 19200

Config Lcd = 16 \* 2

Config Lcdpin = Pin , Db4 = P0.4 , Db5 = P0.5 , Db6 = P0.6 , Db7 = P0.7 , E =  
P0.3 , Rs = P0.2

On Int0 Int0\_isr

Enable Interrupts

Enable Int0

Dim Temp\_flag As Bit

Dim Time\_flag As Bit

Dim Rpm\_flag As Bit

Dim Pb\_flag As Bit

Dim Ti\_flag As Bit

Dim Td\_flag As Bit

Dim Strt\_flag As Bit

Dim Stp\_flag As Bit

Dim Balik As Bit

Dim Ulang As Bit

Dim Keypad As Byte

Dim Iterasi As Byte

Dim Data\_seri As Byte

Dim Temperature As Word

Dim Time As Word

Dim Rpm As Word

Dim Pb As Word

Dim Ti As Word  
Dim Td As Word  
Dim Pvt As String \* 3  
Dim Pvr As String \* 3  
Dim Dop As String \* 3

Led1 Alias P0.0  
Buzzer Alias P0.1

Temperature = 25  
Time = 0  
Rpm = 0  
Program\_utama:  
Reset Led1  
Reset Buzzer  
Time\_flag = 0  
Ti\_flag = 0  
Td\_flag = 0  
Strt\_flag = 0  
Stp\_flag = 0  
Temp\_flag = 0  
Rpm\_flag = 0  
Pb\_flag = 0  
Balik = 1  
Cursor Off  
Cls  
Upperline  
Lcd " PID CONTROLLER "  
Lowerline  
Lcd "V:1.2 FISIKA UI"  
Do  
Loop Until Balik = 0

```
If Ti_flag = 1 Then Goto Ti_prog
If Td_flag = 1 Then Goto Td_prog
If Strt_flag = 1 Then Goto Start_prog
If Time_flag = 1 Then Goto Time_prog
If Pb_flag = 1 Then Goto Pb_prog
If Temp_flag = 1 Then Goto Temp_prog
If Rpm_flag = 1 Then Goto Rpm_prog
If Stp_flag = 1 Then Goto Stop_program
Goto Program_utama
```

```
Int0_isr:
Keypad = P2
Set Led1
Set Buzzer
Waitms 100
Reset Led1
Reset Buzzer
If Keypad > 15 Then
  Keypad = Keypad / 16
  Select Case Keypad
```

```
  Case 1:
    Ti_flag = 1
```

```
  Case 2:
    Td_flag = 1
```

```
  Case 3:
    Strt_flag = 1
```

```
  Case 4:
    Stp_flag = 1
```

```
  Case 5:
    Time_flag = 1
```

```

Case 6:
Pb_flag = 1
Case 7:
Temp_flag = 1
Case 8:
Rpm_flag = 1
End Select
Balik = 0
End If
Return

Start_prog:
Print "*" ; "START:" ; Temperature ; ":" ; Rpm ; ":" ; Time ; "#"
Goto Display_prog

Stop_program:
Print "*" ; "STOP:" ; Temperature ; ":" ; Rpm ; ":" ; Time ; "#"
Goto Program_utama

Temp_prog:
Disable Interrupts
Ulang = 1
Iterasi = 0
Cursor Off
Bitwait P3.2 , Set
Lowerline
Lcd "      "
Temperature = 25
Locate 2 , 1
Lcd "TEMPERATUR=" ; Temperature ; Chr(223) ; "C"
Do
    Bitwait P3.2 , Reset

```

```

Keypad = P2
Set Led1
Set Buzzer
Waitms 100
Reset Led1
Reset Buzzer
If Keypad < 16 Then
    Iterasi = Iterasi + 1
    If Iterasi = 1 Then
        Temperature = Keypad
    Else
        Temperature = 10 * Temperature
        Temperature = Temperature + Keypad
    End If
End If
If Temperature > 400 Then Temperature = 400
Lcd "      "
Locate 2 , 1
Lcd "TEMPERATUR=" ; Temperature ; Chr(223) ; "C"
Bitwait P3.2 , Set
If Keypad > 15 Then
    Keypad = Keypad / 16
    If Keypad = 7 Then
        Ulang = 0
        Print "*" ; "TEMPERATURE:" ; Temperature ; "#"
    End If
End If
Loop Until Ulang = 0
Balik = 1
Enable Interrupts
If Strt_flag = 1 Then
    Temp_flag = 0

```



```
Goto Display_prog
Else
  Goto Program_utama
End If
```

```
Time_prog:
Disable Interrupts
Ulang = 1
Iterasi = 0
Cursor Off
Bitwait P3.2 , Set
Lowerline
Lcd "      "
Time = 0
Locate 2 , 1
Lcd "DURATION=" ; Time ; " Mnt"
Do
  Bitwait P3.2 , Reset
  Keypad = P2
  Set Led1
  Set Buzzer
  Waitms 100
  Reset Led1
  Reset Buzzer
  If Keypad < 16 Then
    Iterasi = Iterasi + 1
    If Iterasi = 1 Then
      Time = Keypad
    Else
      Time = 10 * Time
      Time = Time + Keypad
    End If
  End If
```

```

End If
If Time > 999 Then Time = 999
Lcd "      "
Locate 2 , 1
Lcd "DURATION=" ; Time ; " Mnt"
Bitwait P3.2 , Set
If Keypad > 15 Then
    Keypad = Keypad / 16
    If Keypad = 5 Then
        Ulang = 0
        Print "*" ; "TIME:" ; Time ; "#"
    End If
End If
Loop Until Ulang = 0
Balik = 1
Enable Interrupts
If Strt_flag = 1 Then
    Time_flag = 0
    Goto Display_prog
Else
    Goto Program_utama
End If

Rpm_prog:
Disable Interrupts
Ulang = 1
Iterasi = 0
Cursor Off
Bitwait P3.2 , Set
Lowerline
Lcd "      "
Rpm = 0

```

```

Locate 2 , 1
Lcd "SPEED = " ; Rpm ; " RPM"
Do
  Bitwait P3.2 , Reset
  Keypad = P2
  Set Led1
  Set Buzzer
  Waitms 100
  Reset Led1
  Reset Buzzer
  If Keypad < 16 Then
    Iterasi = Iterasi + 1
    If Iterasi = 1 Then
      Rpm = Keypad
    Else
      Rpm = 10 * Rpm
      Rpm = Rpm + Keypad
    End If
  End If
  If Rpm > 600 Then Rpm = 600
  Lcd "      "
  Locate 2 , 1
  Lcd "SPEED = " ; Rpm ; " RPM"
  Bitwait P3.2 , Set
  If Keypad > 15 Then
    Keypad = Keypad / 16
    If Keypad = 8 Then
      Ulang = 0
      Print "*" ; "RPM:" ; Rpm ; "#"
    End If
  End If
End Do
Loop Until Ulang = 0

```

```

Balik = 1
Enable Interrupts
If Strt_flag = 1 Then
    Rpm_flag = 0
    Goto Display_prog
Else
    Goto Program_utama
End If

```

```

Pb_prog:
If Strt_flag = 0 Then
    Disable Interrupts
    Ulang = 1
    Iterasi = 0
    Pb = 0
    Cursor Off
    Bitwait P3.2 , Set
    Lowerline
    Lcd "      "
    Locate 2 , 1
    Lcd " PB = " ; Pb ; " %"
    Do
        Bitwait P3.2 , Reset
        Keypad = P2
        Set Led1
        Set Buzzer
        Waitms 100
        Reset Led1
        Reset Buzzer
    If Keypad < 16 Then
        Iterasi = Iterasi + 1
        If Iterasi = 1 Then

```

```

        Pb = Keypad
    Else
        Pb = 10 * Pb
        Pb = Pb + Keypad
    End If
End If
Lcd "      "
Locate 2 , 1
Lcd " PB = " ; Pb ; " %"
Bitwait P3.2 , Set
If Keypad > 15 Then
    Keypad = Keypad / 16
    If Keypad = 6 Then
        Ulang = 0
        Print "*P:" ; Pb ; "#"
    End If
End If
Loop Until Ulang = 0
Balik = 1
Enable Interrupts
End If
Goto Program_utama

```

```

Ti_prog:
If Strt_flag = 0 Then
    Disable Interrupts
    Ulang = 1
    Iterasi = 0
    Ti = 0
    Cursor Off
    Bitwait P3.2 , Set
    Lowerline

```

```

Lcd "      "
Locate 2 , 1
Lcd " Ti = " ; Ti ; " mS"
Do
    Bitwait P3.2 , Reset
    Keypad = P2
    Set Led1
    Set Buzzer
    Waitms 100
    Reset Led1
    Reset Buzzer
    If Keypad < 16 Then
        Iterasi = Iterasi + 1
        If Iterasi = 1 Then
            Ti = Keypad
        Else
            Ti = 10 * Ti
            Ti = Ti + Keypad
        End If
    End If
    Lcd "      "
    Locate 2 , 1
    Lcd " Ti = " ; Ti ; " mS"
    Bitwait P3.2 , Set
    If Keypad > 15 Then
        Keypad = Keypad / 16
        If Keypad = 1 Then
            Ulang = 0
            Print "*I:" ; Ti ; "#"
        End If
    End If
Loop Until Ulang = 0

```

```
Balik = 1
Enable Interrupts
End If
Goto Program_utama
```

```
Td_prog:
```

```
If Strt_flag = 0 Then
  Disable Interrupts
  Ulang = 1
  Iterasi = 0
  Td = 0
  Cursor Off
  Bitwait P3.2 , Set
  Lowerline
  Lcd "      "
  Locate 2 , 1
  Lcd " Td = " ; Td ; " mS"
  Do
    Bitwait P3.2 , Reset
    Keypad = P2
    Set Led1
    Set Buzzer
    Waitms 100
    Reset Led1
    Reset Buzzer
  If Keypad < 16 Then
    Iterasi = Iterasi + 1
    If Iterasi = 1 Then
      Td = Keypad
    Else
      Td = 10 * Td
      Td = Td + Keypad
```

```

    End If
End If
Lcd "      "
Locate 2 , 1
Lcd " Td = " ; Td ; " mS"
Bitwait P3.2 , Set
If Keypad > 15 Then
    Keypad = Keypad / 16
    If Keypad = 2 Then
        Ulang = 0
        Print "*D:" ; Td ; "#"
    End If
End If
Loop Until Ulang = 0
Balik = 1
Enable Interrupts
End If
Goto Program_utama

Display_prog:
Cls
Locate 1 , 1
Lcd "SPT=" ; Temperature
Locate 1 , 10
Lcd "PVT=" ; Pvt
Locate 2 , 1
Lcd "RPM=" ; Pvr
Locate 2 , 10
Lcd "DOP=" ; Dop
Do
    Keypad = P2
    If Keypad = 64 Then

```



```

    Stp_flag = 1
End If
If Temp_flag = 1 Then Goto Temp_prog
If Time_flag = 1 Then Goto Time_prog
If Rpm_flag = 1 Then Goto Rpm_prog
Data_seri = Inkey()
If Data_seri = 42 Then
    Pvt = ""
    Ulang = 1
    Do
        Data_seri = Waitkey()
        If Data_seri = 58 Then
            Ulang = 0
        Else
            Pvt = Pvt + Chr(data_seri)
        End If
    Loop Until Ulang = 0
    Pvr = ""
    Ulang = 1
    Do
        Data_seri = Waitkey()
        If Data_seri = 58 Then
            Ulang = 0
        Else
            Pvr = Pvr + Chr(data_seri)
        End If
    Loop Until Ulang = 0
    Dop = ""
    Ulang = 1
    Do
        Data_seri = Waitkey()
        If Data_seri = 35 Then

```

```
    Ulang = 0
Else
    Dop = Dop + Chr(data_seri)
End If
Loop Until Ulang = 0
Locate 1 , 14
Lcd " "
Locate 1 , 14
Lcd Pvt
Locate 2 , 5
Lcd " "
Locate 2 , 5
Lcd Pvr
Locate 2 , 14
Lcd " "
Locate 2 , 14
Lcd Dop
End If
Loop Until Stp_flag = 1
Goto Stop_program
```