



UNIVERSITAS INDONESIA

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

LAPORAN TUGAS AKHIR

NURHIJAH

0606109303

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

DEPARTEMEN FISIKA

PROGRAM DIPLOMA 3 FISIKA INSTRUMENTASI

DEPOK

JULI 2009



UNIVERSITAS INDONESIA

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

LAPORAN TUGAS AKHIR

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Ahli Madya
(A.Md)**

NURHIJAH

0606109303

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

DEPARTEMEN FISIKA

PROGRAM DIPLOMA 3 FISIKA INSTRUMENTASI

DEPOK

JULI 2009

HALAMAN PERNYATAAN ORISINALITAS

**Laporan Tugas Akhir ini adalah hasil karya saya sendiri,
dari semua sumber baik yang dikutip maupun dirujuk
telah saya nyatakan dengan benar.**

Nama : NURHIJAH

NPM : 0606109303

Tanda Tangan :

Tanggal : 9 Juli 2009

HALAMAN PENGESAHAN

Tugas Akhir ini diajukan oleh:

Nama : Nurhijah
NPM : 0606109303
Program Studi : D3 Fisika Instrumentasi Industri
Judul Tugas Akhir : Sistem pengendali Temperatur dan Kecepatan
Shaking Water Bath Berbasis Microcontroller

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar **Diploma D3 pada Program Studi Fisika Instrumentasi Elektronika Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Indonesia**

DEWAN PENGUJI

Pembimbing : Drs. Arief Sudarmaji, M.T. (.....)

Penguji : Dr.Sastra Kusuma.W (.....)

Penguji : Surya Darma,M.si (.....)

Ditetapkan di : Depok

Tanggal : 9 Juli 2009

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 “ Sistem Pengendali Temperatur dan Kecepatan *Shaking Water Bath* Berbasis *Microcontroller*” yang bertujuan untuk memenuhi syarat dalam menyelesaikan pendidikan program studi Diploma 3 Instrumentasi Elektronika dan Industri, 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. Kedua Orang tuaku, Kakek dan Nenekku yang tercinta, yang telah memberi dukungan dan doa yang tak moril dan materiil selama ini.
2. Adikku Lita dan Zulham yang memberikan semangat dan doanya.
3. Drs. Arief Sudarmaji, M.T selaku dosen pembimbing yang telah memberikan petunjuk, kemudahan dalam berpikir dan bimbingan dalam penyelesaian tugas akhir ini.
4. Dr. Prawito selaku Ketua program Diploma 3 Instrumentasi Elektronika dan Industri FMIPA, UI.
5. Surya Darma, M.Si selaku koordinator Tugas Akhir program Diploma 3 Instrumentasi Elektronika dan Industri FMIPA, UI.

6. Dosen-dosen pengajar Jurusan Instrumentasi Elektronika dan Industri yang selama hampir tiga tahun telah memberikan ilmu-ilmunya.
7. Sahabat-sahabatku seperjuangan yang tergabung dalam bersatu dalam TA yang telah memberikan bantuan, dorongan dan doa.
8. Seluruh rekan-rekan Instrumentasi Industri dan Elektronika angkatan 2005
9. Kepada sahabat-sahabatku, shasha, hapsah, uchil, efri, tia, bari, douglas, aziz dwee dll.
10. Seluruh keluarga besar FMIPA UI.
11. Semua pihak yang secara tidak langsung terlibat dalam pembuatan skripsi ini dan tidak mungkin dapat disebutkan satu persatu, semoga amal baik yang telah dilakukan senantiasa dibalas oleh Allah SWT.

Semoga Allah SWT melimpahkan segala rahmat dan karunia-Nya atas kebaikan Bapak / Ibu dan Saudara/i sekalian. Semoga penulisan ilmiah ini benar-benar dapat memberikan kontribusi positif dan menimbulkan sikap kritis kepada para pembaca khususnya dan masyarakat pada umumnya untuk senantiasa terus memperoleh wawasan dan ilmu pengetahuan di bidang teknologi.

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.

Depok, Juli 2009

Penulis

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

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

Nama : Nurhijah
NPM : 0606109303
Program Studi : D3 Instrumentasi Industri
Departemen : Fisika
Fakultas : Matematika dan Ilmu Pengetahuan Alam
Jenis Karya : Laporan Tugas Akhir

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

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

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 : 9 Juli 2009

Yang menyatakan,

(Nurhijah)

ABSTRAK

Nama : Nurhijah

NPM :0606109303

Judul : Sistem Pengendali Temperatur dan Kecepatan Shaking Water Bath
Berbasis *Microcontroller*

Telah dibuat *Shaking Water Bath* Temperatur dan Kecepatan Terkendali dengan menggunakan heater 1500 W, 220 V dengan sensor Termokopel tipe K. Digunakan sebuah receiver (keypad) untuk mengendalikan dan menampilkan nilai dari kecepatan putaran (rpm) dan temperatur (°C) dengan memasukkan nilai *Set Point* (SP). Untuk pengaturan kecepatan putaran dengan cara mengatur lebar pulsa (PWM). Dengan power yang maksimal, temperatur yang dihasilkan hanya 98 °C dengan kecepatan maksimal 600 rpm.

Kata Kunci : Heater, Termokopel, Motor DC, Pengendali.

ABSTRACT

Shaking Water Bath Temperature and speed control have been made with using heater 1500 W, 220 V and thermocouple sensor type K. This heater use a receiver (keypad) for controlling and viewing the valve of rotation speed (rpm) and temperature (°C) by set in set point (SP). Way of rotation speed regulation is with controll the width of pulse (PWM). With maximum power, will have the temperature only 98°C with maximum speed 600 rpm.

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

DAFTAR ISI

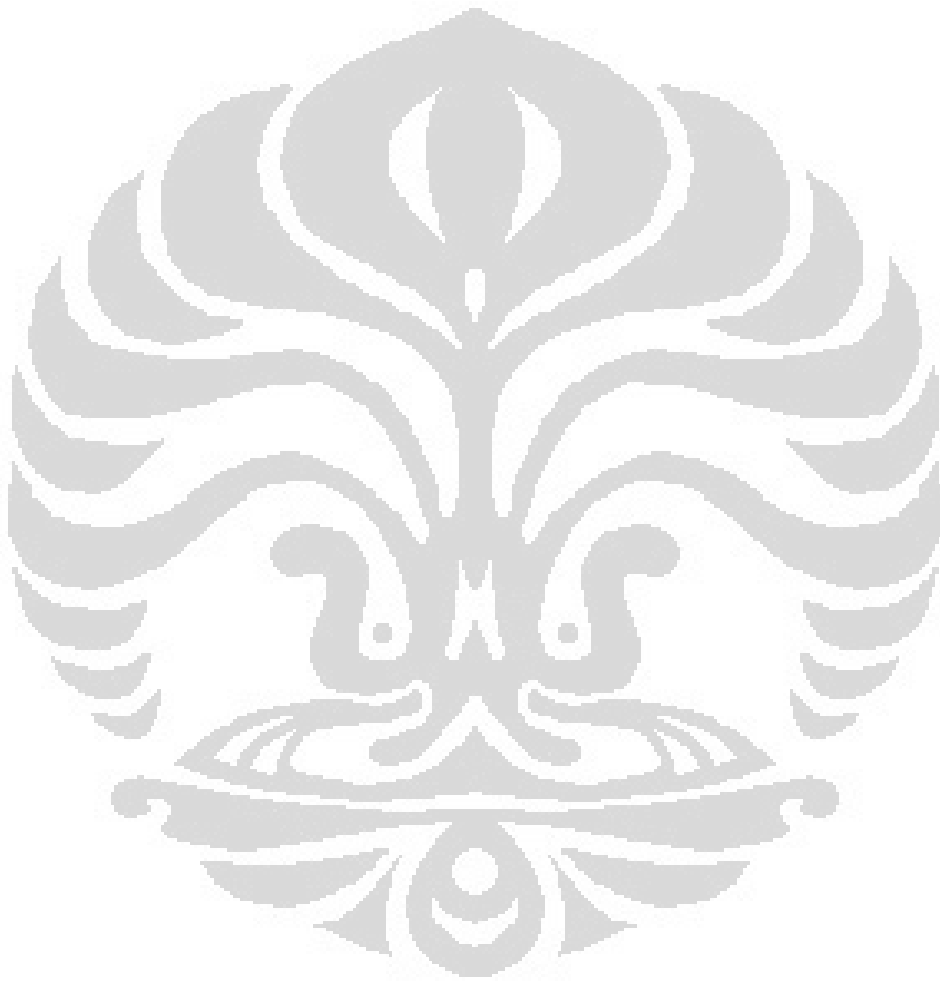
	Halaman
HALAMAN JUDUL.....	i
HALAMAN PERNYATAAN OROSINALITAS	ii
HALAMAN PENGESAHAN.....	iii
KATA PENGANTAR	iv
HALAMAN PERSETUJUAN PUBLIKASI KARYA ILMIAH	vi
ABSTRAK.....	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR	x
DAFTAR TABEL.....	xii
DAFTAR LAMPIRAN.....	.xiii
BAB 1.PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan Penelitian	2
1.3 Deskripsi singkat.....	2
1.4 Batasan Masalah.....	3
1.5 Metodologi Penelitian.....	3
1.6 Sistematika Penulisan	5
BAB 2.TEORI DASAR.....	6
2.1 Heater	6
2.2 Sensor Termokopel	7
2.3 Motor DC	10
2.3.1 Prinsip Kerja Motor DC	10
2.4 PWM (<i>Pulse Width Modulation</i>)	13
2.5 Komunikasi Data Serial RS 232	15
2.6 <i>Solid State Relay</i> (SSR).....	17
2.6.1 Operasi.....	17
2.6.2 Keuntungan SSR.....	17

2.6.3 Kerugian SSR	18
2.7 Teori Kontrol Proporsional Integral Differensial (PID)	18
2.7.1 Teori Dasar Kurva Reaksi dengan Metode Ziegler-nichols	23
2.7.1.1 Keuntungan Pengendalian dengan Metode Ziegler-nichols	24
2.7.1.2 Kerugian Pengendalian dengan Metode Ziegler-nichols.....	24
BAB 3. PERANCANGAN DAN KERJA SISTEM	26
3.1 Perancangan Kerja Sistem	26
3.2 Perancangan Kerja Keras (<i>Hardware</i>)	28
3.2.1 Rangkaian <i>Minimum system</i>	28
3.2.2 Rangkaian <i>Driver Motor</i>	29
3.2.3 Rangkaian <i>Cold Junction</i>	30
3.3 Perancangan Perangkat Lunak(<i>Software</i>).....	31
3.3.1 Proses pada <i>keypad</i>	31
3.3.2 Pengambilan Data.....	31
3.4 Proses pada sistem Minimum Atmega16.....	32
BAB 4. PENGUJIAN SISTEM DAN ANALISA	36
4.1 Pengujian <i>Thermocouple</i>	36
4.2 Pengujian Sistem Pengendali dengan Metode Ziegler-nichols	38
4.3 Hasil Pengujian Sistem Pengendalian dengan $SP=45^{\circ} c$	41
BAB 5. KESIMPULAN DAN SARAN	42
5.1 Kesimpulan	42
5.2 Saran	42
DAFTAR ACUAN	43
LAMPIRAN	

DAFTAR GAMBAR

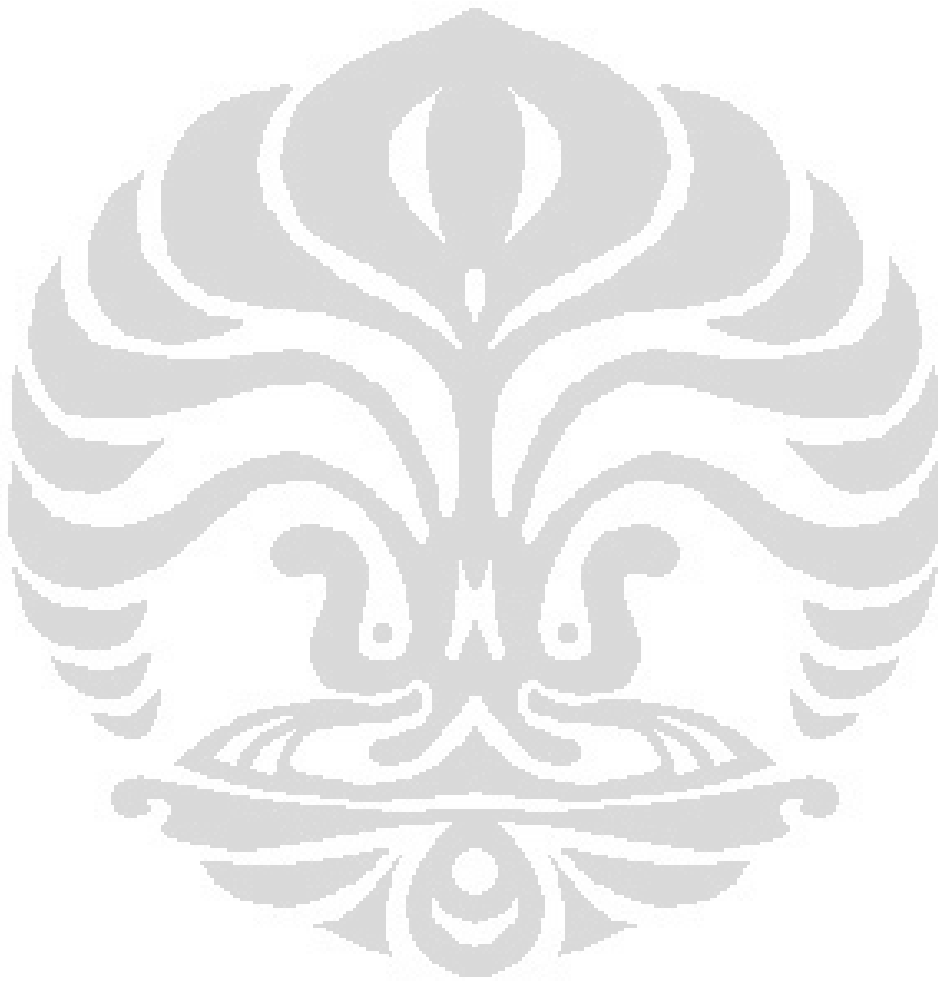
	Halaman
Gambar 1.1 Bagan Keseluruhan sistem	3
Gambar 2.1 Heater	7
Gambar 2.2 Termokopel Berdasarkan Prinsip <i>Seebeck</i>	8
Gambar 2.3 Grafik Respon Termokopel.....	9
Gambar 2.4 Bentuk Fisik Termokopel yang Digunakan	10
Gambar 2.5 Proses Konversi Energi Pada Motor DC.....	11
Gambar 2.6 Motor DC	11
Gambar 2.7 Prinsip Motor DC	12
Gambar 2.8 Gaya Medan Magnet	13
Gambar 2.9 Sinyal PWM Dengan Duty Cycle 50 %	14
Gambar 2.10 Konfigurasi Slot Konektor Serial DB-9	16
Gambar 2.11 Sistem Pengendali Loop terbuka.....	19
Gambar 2.12 Sistem Pengendali Loop Tertutup.....	20
Gambar 2.13 Blok Diagram Pengendali <i>Proportional</i>	20
Gambar 2.14 Grafik respon Pengendali <i>Proportional</i>	21
Gambar 2.15 Blok Diagram Pengendali <i>Integral</i>	21
Gambar 2.16 Grafik Respon Pengendali <i>Integral</i>	22
Gambar 2.17 Blok Diagram Pengendali <i>Differensial</i>	22
Gambar 2.18 Grafik Respon Pengendali <i>Differensial</i>	23
Gambar 2.19 Contoh Respon Pengendali Terhadap Waktu	23
Gambar 3.1 Blok Diagram Cara Kerja Alat.....	26
Gambar 3.2 Blok Diagram Pengendalian Temperature	27
Gambar 3.3 Blok Diagram Pengendalian Motor	28
Gambar 3.4 Minimum system <i>Microcontroller AVR</i>	28
Gambar 3.5 Rangkaian Driver Motor	29
Gambar 3.6 Rangkaian <i>Cold Junction</i>	30
Gambar 3.7 Flowchart program pengendali.....	33
Gambar 4.1 Grafik Persamaan ADC.....	37
Gambar 4.2 Respon waktu proses perubahan temperature	38
Gambar 4.3 Mencari nilai L dan nilai T.....	39

Gambar 4.4 Grafik Pengendalian temperatur dengan SP=45° c 41



DAFTAR TABEL

	Halaman
Tabel 2.1 Jenis Heater dan Kegunaannya	7
Tabel 2.2 Daftar nama pin dan sinyal konektor serial DB-9.....	17
Tabel 2.3 Data ADC.....	37

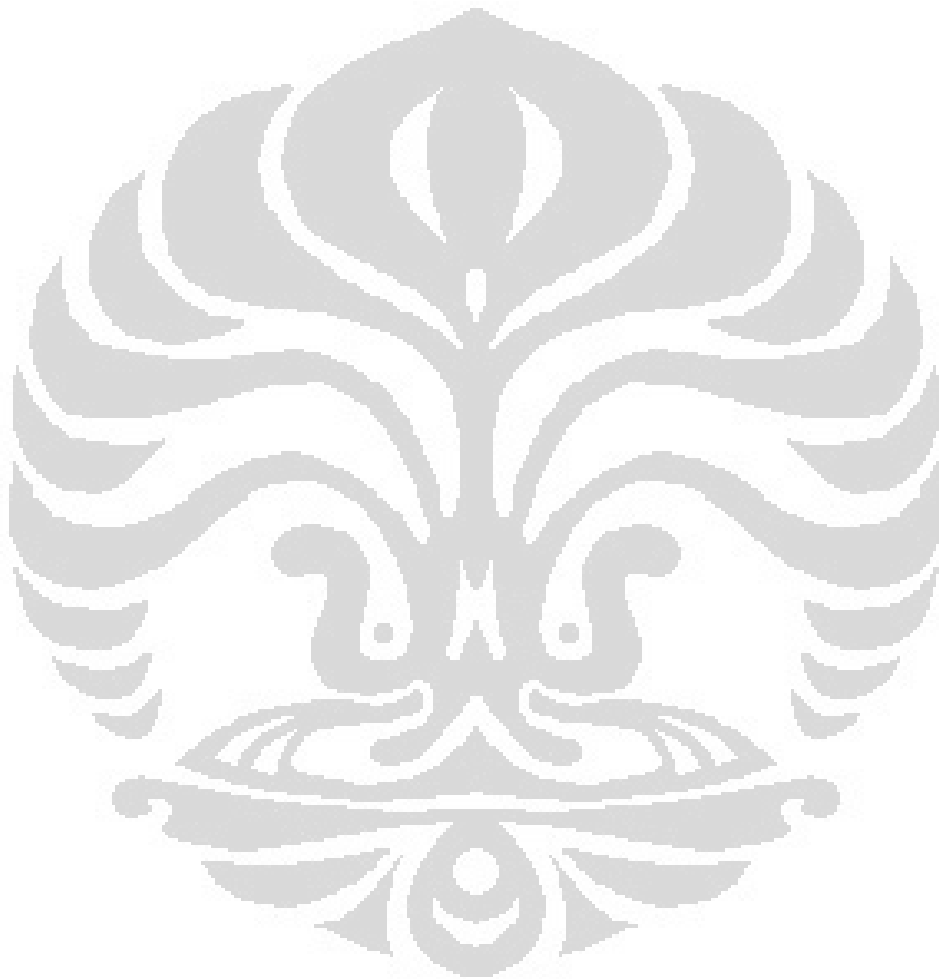


DAFTAR LAMPIRAN

Lampiran 1 : Rangkaian *Minsys*

Lampiran 2 : Program Pengambilan Data

Lampiran 3 : Program Utama



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

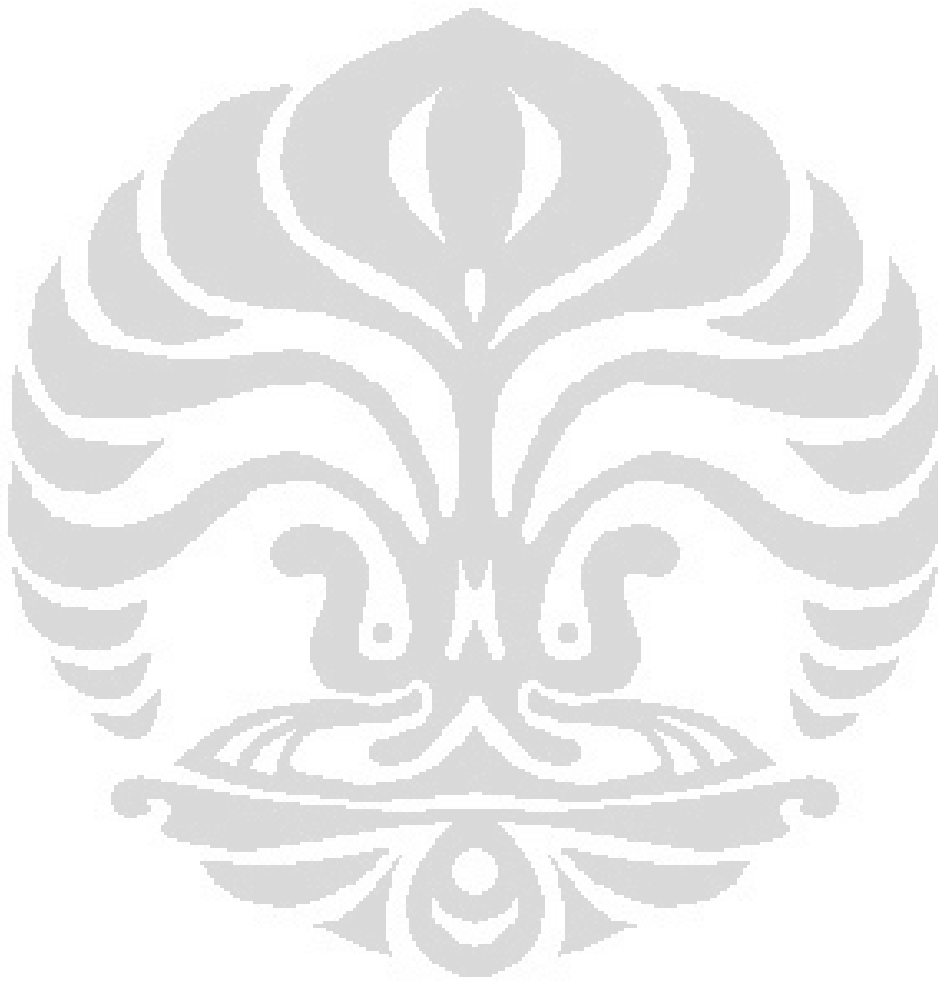
Perkembangan dunia teknologi menghasilkan berbagai macam cara untuk memudahkan kerja manusia. Tidak hanya proses industri yang lekat dengan teknologi. Dunia pendidikan, pertahanan, pertanian, bahkan kesehatan sudah memanfaatkan teknologi untuk mempermudah dan mengefisiensikan waktu kerja.

Untuk memenuhi kebutuhan akan informasi teknologi, banyak literatur dan katalog produk yang tersebar dengan maksud memperkenalkan dan mengajak masyarakat ataupun instansi untuk menikmati teknologi. Karena kebutuhan yang tinggi, jumlah produsen teknologi yang terbatas, barang-barang teknologi yang dijual dengan harga yang sangat tinggi. Oleh sebab itu penulis mencoba untuk membuat sebuah alat yang terdapat pada katalog produk kesehatan dengan harga hemat. Hal ini dilakukan untuk membuktikan bahwa untuk menggunakan teknologi, kita tidak harus mengeluarkan biaya yang cukup besar.

Alat ini membantu dunia kesehatan, dimana sistem kerjanya berupa pengendalian kecepatan dan temperatur. Pengendalian dilakukan dengan *keypad* yang kemudian diterjemahkan *microcontroller* dan selanjutnya di kirim ke *actuator* untuk diproses agar didapat hasil yang kita inginkan. Sehingga kerjanya sangat membantu mengefisiensikan waktu dan tenaga pada dunia kesehatan yang terbilang sibuk. Perbedaan biaya produksi dengan harga jual alat ini sangatlah besar. Sehingga dapat mengurangi beban pembelian pada dunia kesehatan yang selama ini cukup menguras biaya.

1.2 Tujuan Penelitian

Penulisan Tugas Akhir ini merupakan salah satu syarat dalam menyelesaikan kurikulum Program D3 Fisika Instrumentasi dengan Jurusan Instrumentasi Elektronika, Departemen Fisika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Indonesia.



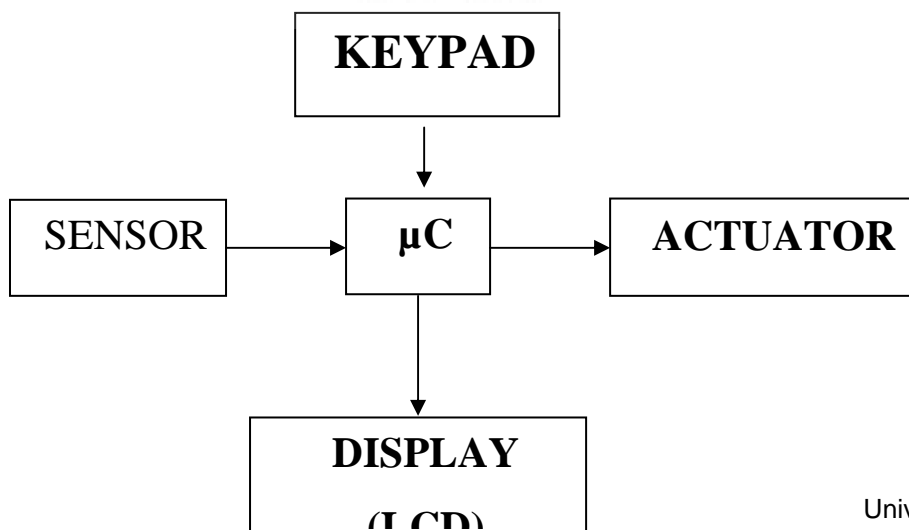
1.3 Deskripsi Singkat

Fungsi alat-alat yang dihasilkan oleh dunia teknologi tidak hanya terbatas pada proses industri. Pada penelitian dibidang kesehatan pun sangat terbantu dengan keberadaan teknologi yang semakin pesat.

Sebagai salah satu aplikasi teknologi pada penelitian dibidang kesehatan, pada tugas akhir ini dibuat “**Sistem Pengendali Temperatur dan Kecepatan *shaking water bath* Berbasis Microcontroler**” dengan maksud menekan pembiayaan dan memudahkan pekerja dalam bidang penelitian sehingga proses kerja yang dilakukan dapat berjalan dengan mudah, aman dan efisien. Hal tersebut dikarenakan pengontrolan kerja untuk kenaikan temperatur alat yang dilakukan dengan *keypad*.

Cara kerja alat ini adalah pada saat kita ingin menghasilkan temperatur yang cukup tinggi, maka tombol temperatur ditekan untuk diatur sesuai dengan keinginan sehingga terjadi proses pemanasan, yang sebelumnya telah kita atur 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 pengaduk, dan *heater* 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

Pada pembuatan alat penulis membagi menjadi dua bagian pengerjaan, bagian yang pertama mengenai sistem mekanik mengenai rancang bangun dari alat tersebut beserta rangkaiannya dan bagian yang kedua mengenai sistem pengendalian dengan menggunakan program yang berbasis *microcontroller*.

Dalam hal ini penulis memfokuskan mengenai sistem pengendali dengan menggunakan program berbasis *microcontroller* yang digunakan untuk keranjang dan motor DC (*actuator*). Pengendalian tersebut dilakukan oleh *keypad* yang kemudian diterjemahkan oleh *micro* untuk menggerakkan *actuator* sesuai dengan perintah yang diinginkan dan *display* menterjemahkannya dalam tulisan. Sensor pada alat ini untuk membaca temperatur pada alat.

1.5 Metodologi 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 perencanaan alat berupa sistem pengendali berbasis *microcontroller* dan mekanik. Pada bagian mekanik akan membahas desain dan cara kerjanya, serta sensor. Pada bagian sistem pengendali akan membahas masalah pembuatan minimum sistem *microcontroller*, *controller* dan hot plate.

3. Pembuatan Program

Tahap ini merupakan proses pembuatan program dilakukan dengan menggunakan *Software Basic Compiler (BASCOS)*, 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 keranjang.

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 pembuatan sebuah alat dibutuhkan landasan-landasan teori tentang alat tersebut dan akan digunakan dalam proses pembuatan alat tersebut. Pada bab ini akan dijelaskan landasan-landasan teori pada alat *shaking water bath* sebagai hasil dari studi literatur yang berhubungan dalam perancangan dan pembuatan alat, antara lain prinsip dasar heater , sensor temperatur, motor DC, PWM (*Pulse Width Modulation*), komunikasi data serial dan SSR (*Solid State Relay*).

2.1 Heater

Heater adalah salah satu komponen utama atau benda temperatur tinggi yang mentransfer energi ke benda yang bertemperatur rendah. Panjang gelombang dari radiasi

infra red dengan range dari 78 nm sampai 1mm. Sebuah klarifikasi dari heater adalah terhubungnya daerah panjang gelombang dari emisi energi utama. Gelombang pendek untuk range dari 78 nm sampai 1400 nm, medium untuk range antara 1400 nm sampai 8000 nm sedangkan untuk range terjauh adalah diatas 3000 nm.

Element dari heter jenis ini kebanyakan memakai konstruksi dari sebuah emisi api atau filament elektrik panas sebaai pemancarnya. Jika sebuah operasi elektrik infra red heater digunakan, filament biasanya akan melindungi dari sebuah resistansi panas. Material yang banyak digunakan pada heater jenis ini adala kawat yang berpelindung, alternatif temperatur rendah dari kawat adalah karbon, campuran dari besi, khromium, aluminium. Industri infrared heater terkadang menggunakan sebuah pelapis emas dalam pipa kwarsa. Emas digunakan karena hambatan oksidasinya sangat tinggi mencapai 95%. Berikut ini adalah table jenis-jenis heater.



Gambar 2.1 Heater

Tabel 2.1 Jenis-jenis heater

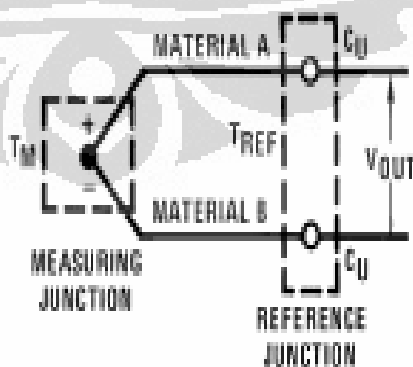
Jenis Heater	Sifat Benda yang Dipanaskan	Memanaskan / Membuat
Tubular Straight, Multiform	Padat	Direkatkan pada dies, heat sealing tools, dll.
Tubular Straight, Multiform	Cair	Air, minyak, plating, aspal, garam, dll
Tubular	Permukaan benda Padat	Drying, baking, kain, plastic, makanan, dll.
Immersion 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.

2.2 Sensor Termokopel

Sensor yang digunakan penulis dalam mengkonversi perbedaan panas dalam benda yang diukur temperaturnya menjadi perubahan potensial atau tegangan listrik adalah termokopel. Termokopel yang digunakan berbentuk sederhana sehingga dapat dipasang dengan mudah, memiliki jenis konektor standar yang sama, dan mampu mengukur temperatur dalam jangkauan suhu yang cukup besar.

Berdasarkan prinsip *seebeck* termokopel merupakan sensor temperatur termoelektrik yang terdiri dari sepasang kawat logam konduktor berbeda jenis dihubungkan bersama-sama pada satu ujung 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 2.2 Termokopel berdasarkan prinsip *Seebeck*

Hubungan antara perbedaan suhu dengan tegangan yang dihasilkan termokopel bukan merupakan fungsi linier melainkan fungsi interpolasi polinomial. Agar diperoleh hasil pengukuran yang akurat, persamaan biasanya diimplementasikan pada kontroler digital.

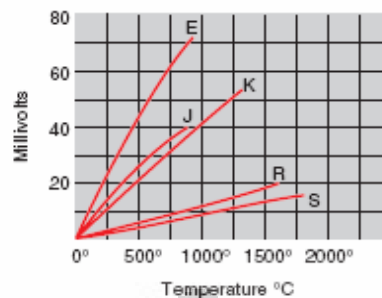
Berdasarkan penggunaannya tersedia beberapa jenis termokopel [4], yaitu :

- ★ Tipe K (Chromel (Ni-Cr alloy) / alumel (Ni-Al alloy)) termokopel ini digunakan untuk temperatur tinggi kira-kira -200°C hingga $+1200^{\circ}\text{C}$, dan harganya pun lebih murah dibanding tipe yang lain.
- ★ Tipe E ((Chromel / Constantan (Cu-Ni alloy)), jenis ini memiliki output yang besar ($68\mu\text{V}/^{\circ}\text{C}$), digunakan untuk temperatur rendah.
- ★ Tipe J (iron / Constantan) rentangnya terbatas (-40 hingga $+750^{\circ}\text{C}$), jenis ini memiliki sensitivitas sekitar $\sim 52\mu\text{V}/^{\circ}\text{C}$.
- ★ Tipe N (Nicrosil (Ni-Cr-Si alloy) / Nisil (Ni-Si alloy)) stabil dan tahan yang tinggi terhadap oksidasi membuat tipe N cocok untuk pengukuran suhu yang tinggi tanpa platinum. Dapat mengukur suhu di atas 1200°C . Sensitivitasnya sekitar $39\mu\text{V}/^{\circ}\text{C}$, sedikit di bawah tipe K.
- ★ Tipe B (Platinum-Rhodium/Pt-Rh) cocok untuk mengukur suhu diatas 1800°C .
- ★ Tipe R (Platinum/Platinum with 7% Rhodium) cocok untuk mengukur suhu di atas 1600°C , sensitivitasnya rendah ($10\mu\text{V}/^{\circ}\text{C}$), termokopel jenis ini jarang sekali dipakai karena harganya yang terlalu mahal.
- ★ Tipe S (Platinum/Platinum with 10% Rhodium) cocok untuk mengukur suhu dia atas 1600°C , sensitivitasnya rendah ($10\mu\text{V}/^{\circ}\text{C}$), jenis termokopel ini pun harganya mahal.
- ★ Tipe T (Copper/Constantan) cocok untuk pengukuran antara -200 to 350°C , jenis ini memiliki sensitivitas $\sim 43\mu\text{V}/^{\circ}\text{C}$.

Dalam pembuatan alat ini menggunakan termokopel tipe K, dimana termokopel tipe 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°C adalah $40,6\mu\text{V}/^{\circ}\text{C}$

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



Gambar 2.3 Grafik Respon Thermocouple



Gambar 2.4 Bentuk Fisik Thermocouple yang Digunakan

2.3 Motor DC

Motor DC berfungsi mengubah energi listrik menjadi energi mekanis yang berupa putaran motor. Motor DC adalah motor yang menggunakan arus searah (*Direct Current*). Motor DC banyak digunakan di industri yang memerlukan presisi tinggi dalam gerak untuk pengaturan kecepatan pada torsi yang konstan. Salah satu komponen yang diperlukan dalam sistem pengendalian adalah aktuator, yaitu komponen pertama untuk melakukan gerakan dengan mengubah energi elektrik menjadi gerakan mekanik. Salah satu jenis aktuator adalah motor listrik.

2.3.1 Prinsip Kerja Motor DC

Motor DC merupakan motor yang memerlukan suplay tegangan searah pada kumparan jangkar dan kumparan medan. Dimana suplay tersebut digunakan untuk diubah menjadi energi mekanik. Kumparan medan pada motor DC yang dialiri arus listrik akan menghasilkan medan magnet yang melingkupi kumparan jangkar dengan arah tertentu.

Mengkonverter energi dari energi listrik menjadi energi mekanik (motor) maupun sebaliknya dari energi mekanik menjadi energi listrik (generator), dimana proses itu berlangsung melalui medium medan magnet. Energi yang akan diubah dari suatu sistem ke sistem yang lain, sementara akan tersimpan pada medium medan magnet yang kemudian dilepaskan menjadi energi system lainnya. Dengan demikian, medan magnet disini selain berfungsi sebagai tempat penyimpanan energi juga sekaligus proses perubahan energi. Proses perubahan energi pada motor arus searah dapat digambarkan pada Gambar 2.5.



Gambar 2.5 Proses Konversi Energi Pada Motor DC

Motor arus searah, sebagaimana namanya, menggunakan arus langsung yang tidak langsung/direct-unidirectional. Motor DC digunakan pada penggunaan khusus dimana diperlukan penyalaan torque yang tinggi atau percepatan yang tetap untuk kisaran kecepatan yang luas.

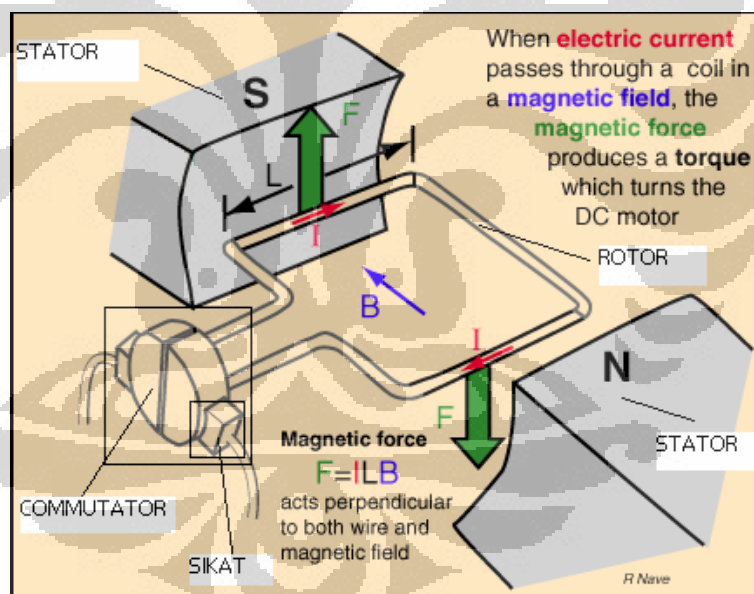


Gambar 2.6 Motor DC

Dibawah ini merupakan enam komponen utama pada sebuah motor DC :

- **Rotor**, yaitu bagian yang berputar pada motor berupa kumparan kawat.
- **Stator**, yaitu bagian pada motor berupa magnet.
- **Commutator**, komponen ini terutama ditemukan dalam motor DC. Kegunaannya adalah untuk membalikan arah arus listrik dalam dinamo. *Commutator* juga membantu dalam transmisi arus antara dinamo dan sumber daya.
- **Sikat**, yaitu sepasang batang garfit yang menempel pada comutator tetapi tidak berputar

- **Kutub medan**, secara sederhana digambarkan bahwa interaksi dua kutub magnet akan menyebabkan perputaran pada motor DC. Motor DC memiliki kutub medan yang stasioner dan dinamo yang menggerakkan bearing pada ruang diantara kutub medan. Motor DC sederhana memiliki dua kutub medan: kutub utara dan kutub selatan. Garis magnetik energi membesar melintasi bukaan diantara kutub-kutub dari utara ke selatan. Untuk motor yang lebih besar atau lebih kompleks terdapat satu atau lebih elektromagnet. Elektromagnet menerima listrik dari sumber daya dari luar sebagai penyedia struktur medan.
- **Dinamo**, bila arus masuk menuju dinamo, maka arus ini akan menjadi elektromagnet. Dinamo yang berbentuk silinder, dihubungkan ke as penggerak untuk menggerakkan beban. Untuk kasus motor DC yang kecil, dinamo berputar dalam medan magnet yang dibentuk oleh kutub-kutub, sampai kutub utara dan selatan magnet berganti lokasi. Jika hal ini terjadi, arusnya berbalik untuk merubah kutub-kutub utara dan selatan dinamo.



Gambar 2.7 Prinsip Motor DC

Seperti terlihat pada Gambar 2.7, motor DC memiliki prinsip kerja yaitu suatu penghantar yang berarus listrik dan ditempatkan dalam suatu medan magnet maka penghantar tersebut akan mengalami gaya. Prinsip kerja motor membutuhkan :

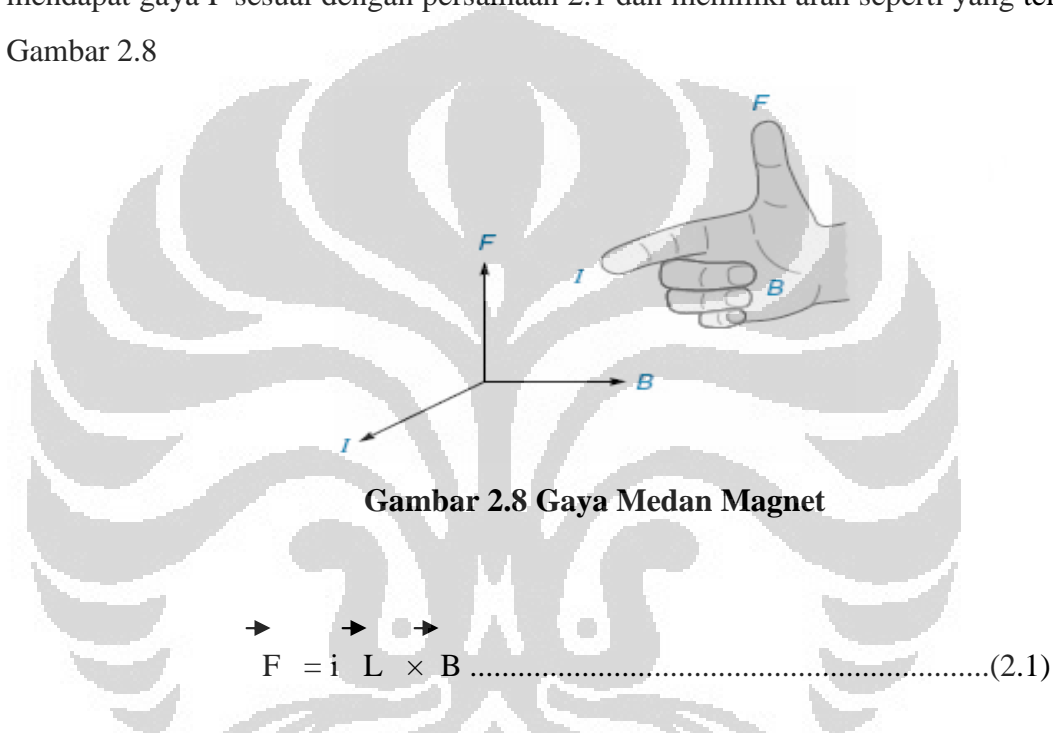
1. Adanya garis-garis gaya medan magnet (fluks), antara kutub yang berada di stator.
2. Penghantar yang berarus listrik yang ditempatkan dalam medan magnet tersebut.

3. Pada penghantar akan timbul gaya.

Gaya yang dihasilkan motor dc tergantung pada :

 - a. Kekuatan pada medan magnet.
 - b. Besarnya arus yang mengalir pada penghantar.
 - c. Panjang kawat penghantar yang berada dalam medan magnet.

Apabila panjang kumparan rotor L dialiri arus listrik sebesar I dan terletak diantara kutub magnet utara dan selatan dengan kerapatan fluks sebesar B , maka kumparan rotor tersebut mendapat gaya F sesuai dengan persamaan 2.1 dan memiliki arah seperti yang terlihat pada Gambar 2.8



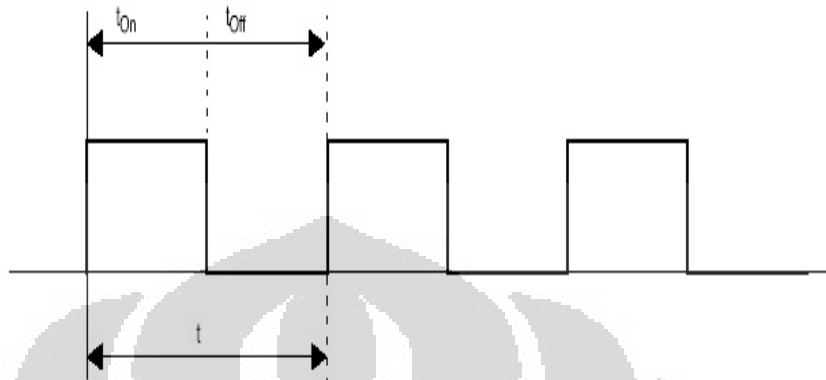
Keterangan :

- F = Gaya Lorentz (Newton)
- B = Kerapatan Fliks Magnet (Weber / m²)
- I = Arus Listrik (Ampere)
- L = Panjang sisi kumparan rotor (m)

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.[6]
 Pemodulasian sinyal yang beragam dapat menghasilkan duty cycle yang diinginkan.
 Gambar 2.9 memperlihatkan sinyal kotak dengan duty cycle 50%.



Gambar 2.9 Sinyal PWM dengan duty cycle 50 %

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 :

$$D = \frac{t_1}{t_1 + t_2} \dots\dots\dots(2.2)$$

Dimana : D = Duty Cycle

$$t_1 = t_{ON}$$

$$t_2 = t_{OFF}$$

sehingga frekuensi yang dapat dihasilkan :

$$F = \frac{1}{T} \quad \text{dimana : } T = \text{periode } (t_1 + t_2) \dots\dots\dots(2.3)$$

Modulasi lebar pulsa yang dihasilkan dari teknik PWM ini akan digunakan untuk mengatur kecepatan dari motor DC. Dari Gambar 2.9 maka kita dapat mengetahui nilai RMS nya dengan menggunakan persamaan (2.6) dimana :

$$V(t) = \begin{cases} 0 \rightarrow t \Rightarrow V_p \\ t \rightarrow T \Leftarrow 0 \end{cases}$$

$$V_{RMS} = \sqrt{\frac{\int_0^{t_1} V(t)^2 dt}{T}}$$

$$V_{RMS} = \sqrt{\frac{\int_0^{t_1} Vp^2 dt + \int_{t_1}^t 0 dt}{T}}$$

$$V_{RMS} = \sqrt{\frac{\int_0^t Vp^2 dt}{T}}$$

$$V_{RMS} = \sqrt{\frac{Vp^2 t}{T}}$$

$$V_{RMS} = \sqrt{\frac{Vp^2 t}{T}}$$

$$V_{RMS} = Vp \sqrt{\frac{t}{T}}$$

$$P = \frac{V_{rms}^2}{R}$$

$$P = \frac{V_P^2(t)}{R (T)} \dots\dots\dots(2.4)$$

2.5 Komunikasi data serial RS 232

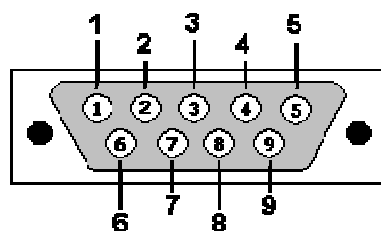
Komunikasi serial adalah pengiriman data secara serial (data dikirim satu persatu secara berurutan) sehingga komunikasi serial jauh lebih lambat daripada komunikasi yang paralel. Karena peralatan berkomunikasi menggunakan transmisi serial sedangkan data dikomputer diolah secara paralel, oleh karena itu harus dikonversikan dahulu ke bentuk

paralel. Jika menggunakan perangkat keras hal ini bisa dilakukan oleh Universal Asynchronous Receiver Transmitter (UART), yang membutuhkan perangkat untuk menanganinya. Komunikasi serial merupakan salah satu cara untuk mengkomunikasikan data dari suatu peralatan ke peralatan lain dengan cara menggunakan data 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 0 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 pewaktu (timing).

Perangkat keras pada komunikasi serial dibagi menjadi dua kelompok, yaitu Data communication Equipment (DCE) dan Data Terminal Equipment (DTE). Modem salah satu contoh DCE, sedangkan terminal yang terdapat di komputer salah satu contoh dari DTE. Null modem digunakan untuk menghubungkan dua buah data DTE, hal ini biasanya dilakukan untuk mentransfer file antar komputer .

Komunikasi serial RS232 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 RS232 pada komputer harus memenuhi standar RS232. Agar level tegangan sesuai dengan tegangan TTL/CMOS diperlukan RS232 level konverter. IC yang banyak digunakan untuk ini adalah MAX-232. Konfigurasi slot DB-9 *female* adalah sebagai berikut :

RS232 DB9 (EIA/TIA 574)



Gambar 2.10 Konfigurasi slot konektor serial DB-9

Fungsi dari masing-masing pin dan sinyal konektor serial B-9 dapat dilihat pada Tabel 2.2 berikut ini:

Tabel 2.2 Daftar nama pin dan sinyal konektor serial DB-9

Pin DB-9	Nama Pin	Keterangan
1	<i>DCD</i>	<i>Data Carrier DetectData Port (DP0 - DP9)</i>
2	<i>RD</i>	<i>Receive Data (a.k.a RxD, Rx)</i>
3	<i>TD</i>	<i>Transmit Data (a.k.a TxD, Tx)</i>
4	<i>DTR</i>	<i>Data Terminal Ready</i>
5	<i>SGND</i>	<i>Ground</i>
6	<i>DSR</i>	<i>Data Set Ready</i>
7	<i>RTS</i>	<i>Request To Send</i>
8	<i>CTS</i>	<i>Clear To Send</i>
9	<i>RI</i>	<i>Ring Indicator</i>

2.6 Solid State Relay (SSR)

SSR adalah sebuah saklar elektrik.tidak seperti saklar elektromekanik, SSR tidak memiliki bagian yang bergerak. Ada beberapa tipe SSR yaitu *photo coupled SSR*, *transformer coupled SSR* dan *hybrid SSR*. *Photo coupled SSR* dikendalikan oleh sinyal tegangan *low*.

2.6.1 Operasi

Tegangan yang digunakan pada rangkaian SSR mengakibatkan *LED* menyinari *photo-sensitive dioda*. Hal ini akan menghasilkan tegangan diantara *MOSFET* dengan *gate* dan mengakibatkan *MOSFET* daalam kondisi *on*. SSR terdiri dari *MOSFET* tunggal atau yang terdiri dari beberapa *MOSFET*.

2.6.2 Keuntungan SSR

- *SSR* Lebih cepat dari saklar elektromekanik. Waktu untuk perubahan kondisi tergantung waktu yang diinginkan
- Lebih awet, karena tidak ada bagian yang bergerak secara mekanik.
- Lebih bersih.
- Mengurangi noise elektrik ketika berubah kondisi.
- Dapat digunakan pada lingkungan yang tidak boleh terjadi bunga api.
- Sunyi dalam perubahan kondisi.
- Lebih kecil dari saklar mekanik yang saling berhubungan.

2.6.3 Kerugian *SSR*

- Lebih mudah rusak ketika terjadi hubungan pendek.
- Menambah noise elektrik ketika terjadi konduktansi.
- Sewaktu kondisi *close*, impedansi lebih besar akibatnya akan menghasilkan panas.
- Sewaktu kondisi *open*, impedansinya lebih kecil.
- Terjadi kebocoran arus balik sewaktu kondisi *open*.
- Kemungkinan adanya kegagalan berubah kondisi ketika waktu tegangannya singkat.
- Lebih mahal harganya dibandingkan saklar elektromekanik.

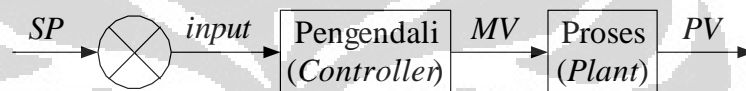
2.7 Teori Kontrol Proporsional Integral Differensial (PID)

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*. 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 aktuatur (contoh: *control valve*).

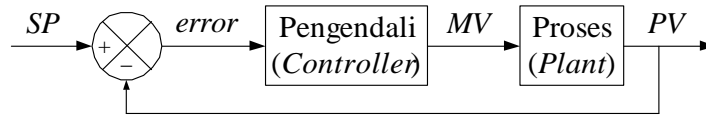
- PV (*Process Variable*) adalah sinyal hasil pemantauan terhadap proses atau *plant*. *Process Variable* umumnya adalah hasil pembacaan dari suatu sensor (contoh: *thermocouple*).
- *Plant* adalah objek yang akan dikendalikan (contoh: temperatur).

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 2.11 menggambarkan sistem pengendali *loop* terbuka (*Open Loop Control System*).



Gambar 2.11 Sistem Pengendali Loop Terbuka

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

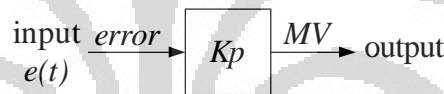


Gambar 2.12 Sistem Pengendali Loop Tertutup

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.



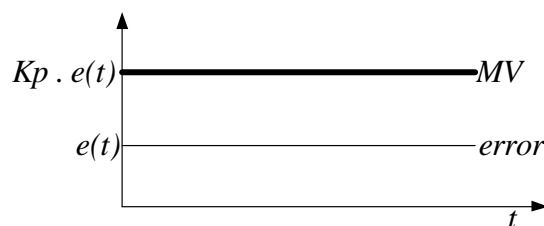
Gambar 2.13 Blok Diagram Pengendali *Proportional*

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

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

$$MV = K_p \cdot e(t) \dots\dots\dots(2.3)$$

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



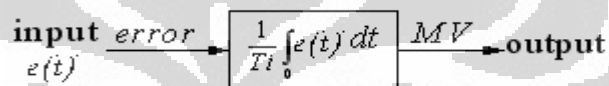
Gambar 2.14 Grafik respon Pengendali *Proportional*

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.

b. Pengendali *Integral* (I)

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

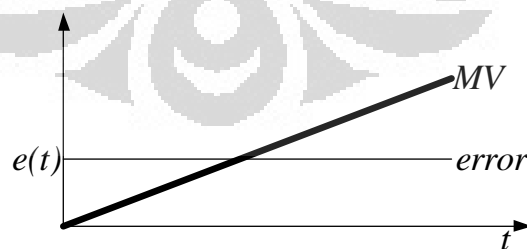


Gambar 2.15 Blok Diagram Pengendali *Integral*

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

$$MV = \frac{1}{T_i} \int_0^t e(t) dt \dots\dots\dots(2.4)$$

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*

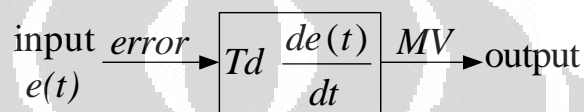
Pengendali *integral* berfungsi untuk mengurangi dan menghilangkan *steady-state error* yang timbul setelah respon *plant* dari pengendali *proportional* sudah stabil. Semakin kecil nilai *steady-state error*, maka respon dari *plant* akan semakin mendekati keadaan

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*.

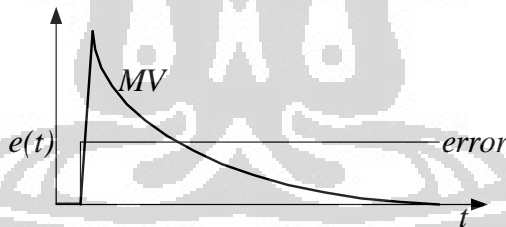
c. Pengendali *Diferensial* (D)

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*

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.



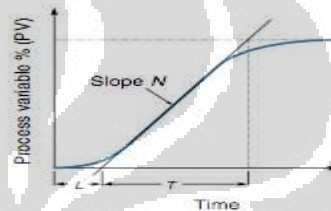
Gambar 2.18 Grafik Respon Pengendali *Diferensial*

Pengendali *diferensial* berfungsi untuk mengurangi respon yang terlalu berlebih yang dapat mengakibatkan *over shoot* pada proses *plant* karena nilai *Kp* yang terlalu besar pada pengendali *proportional*. Output dari pengendali *diferensial* 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 *Kp* pada pengendali *proportional*. Pengendali *diferensial*

hanya akan bekerja ketika terjadi perubahan *error*, sehingga ketika proses yang dikendalikan sudah stabil maka pengendali *differensial* sudah tidak bekerja lagi.

2.7.1 Teori Dasar Kurva Reaksi dengan Metode Ziegler- Nichols

Metode kurva reaksi adalah salah satu cara pengendalian parameter *PID*. Metode ini tidak membutuhkan sistim untuk beresilasi. Sebagai alternatif, *loop* pada *feedback* terbuka dan pengendalian terhadap *output* dilakukan manual secara langsung.



Gambar 2.19 Contoh respon pengendalian terhadap waktu

Dengan :

L = Lag Time

T= Rise time

Dalam melakukan pengendalian dengan kurva reaksi metode Ziegler- Nichols dapat dilakukan melalui beberapa langkah, yaitu:

Mencari persamaan tangen pada bagian raising dari kurva respon. Garis ini akan menghasilkan nilai L (delay) dan T (Rise Time). L merupakan delay waktu antara outputpengendali dengan respon yang dikendalikan.

Menghitung slope dari kurva dengan cara:

$$N = S / \Delta P$$

Dimana :

N = Kemiringan garis singgung yang melalui titik infleksi/ perubahan

ΔP = Besarnya perubahan dalam pengendali output

Menghitung Konstanta *PID* :

$$K_c = 1.2 / L N$$

$$T_i = 2 L$$

$$T_d = 0.5 L$$

II.7.1.1 Keuntungan Pengendalian Kurva Reaksi Metode Ziegler – Nichols

1. Hanya membutuhkan 1 kali eksperimen.
2. Tidak membutuhkan prosedur *trial* dan *error*.
3. Setting kontrolnya lebih mudah diperhitungkan.

II.7.1.2 Kerugian Pengendalian Kurva Reaksi Metode Ziegler – Nichols

1. Eksperimennya dijalankan berdasarkan kondisi open- loop
 - › Gangguan yang terjadi harus dieliminasi selama eksperimen.
2. Titik perubahannya tidak mudah didapat jika pengukuran mengalami gangguan dan grafik pencatatannya tidak cukup besar.
3. Metode ini cenderung sensitif terhadap kesalahan kalibrasi controller. Sedangkan metode Z-N tidak begitu sensitif terhadap kesalahan kalibrasi karena daya controller disesuaikan selama eksperimen.
4. Metode ini tidak direkomendasikan untuk proses yang memiliki respon osilasi.

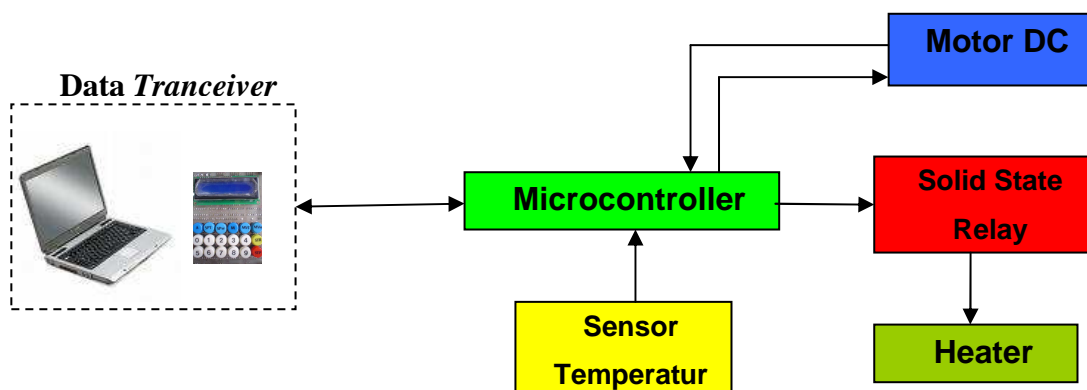
BAB 3

PERANCANGAN DAN CARA KERJA SISTEM

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

3.1 Perancangan Kerja Sistem

Alat pengocok dan pemanas ini dirancang agar dapat mengendalikan temperatur dan kecepatan pergerakan pengaduk sesuai dengan yang diinginkan. Berikut ini adalah cara kerja sistem secara umum:



Gambar 3.1 Blok Diagram Cara Kerja Alat

Dari blok diagram diatas terdapat keypad sebagai data *tranceiver* untuk mengirim nilai *Set Point* (SP) yang berupa bilangan ke dalam *microcontroller* dan menampilkan nilai *Process Variable* (PV) pada LCD. Proses pengendalian motor dan temperatur terjadi didalam *microcontroller*. Kemudian heater dan motor DC bekerja sesuai dengan nilai *Set Point* yang telah diatur. Hasil pembacaan dari sensor temperatur dan sensor shaft encouder dikirimkan ke *microcontroller*. Dan kemudian ditampilkan pada LCD.

Dari blok diagram di atas terdapat keypad sebagai data *tranceiver* untuk mengirimkan nilai *Set Point* (SP) yang berupa bilangan ke dalam *microcontroller* dan menampilkan nilai *Process Variable* (PV). Proses pengendalian temperturnya terjadi di dalam *microcontroller*.

Pada dasarnya, alat ini dikendalikan oleh sistem pengendali yaitu pengendali temperatur. Sistem pengendali yang digunakan untuk pengendali temperturnya adalah *PI controller* (*Proportional Integrator*). Dalam pengendali temperatur, telah ditentukan nilai SP (*Set Point*) berupa bilangan lalu dihitung nilai errornya dengan rumus sebagai berikut:

$$E = SP - PV$$

Dengan:

E = Error

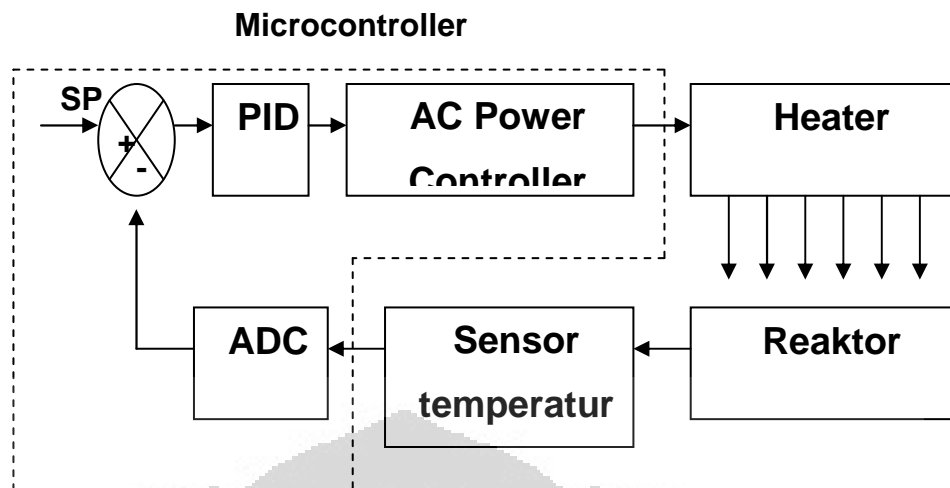
SP = Set Point

PV = Process Variable

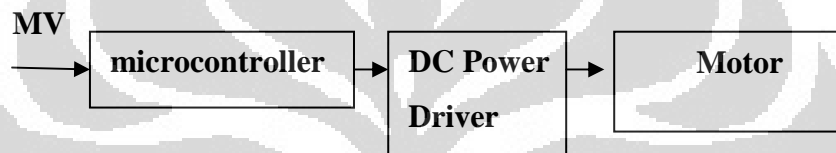
lalu ke dalam PID yang akan dihitung nilai *Manipulated Variable* (MV) dengan persamaan:

$$MV = Kp (E + Ki \int Edt + Kd \frac{dE}{dt}) \dots\dots\dots(3.1)$$

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 Pengendalian Temperature



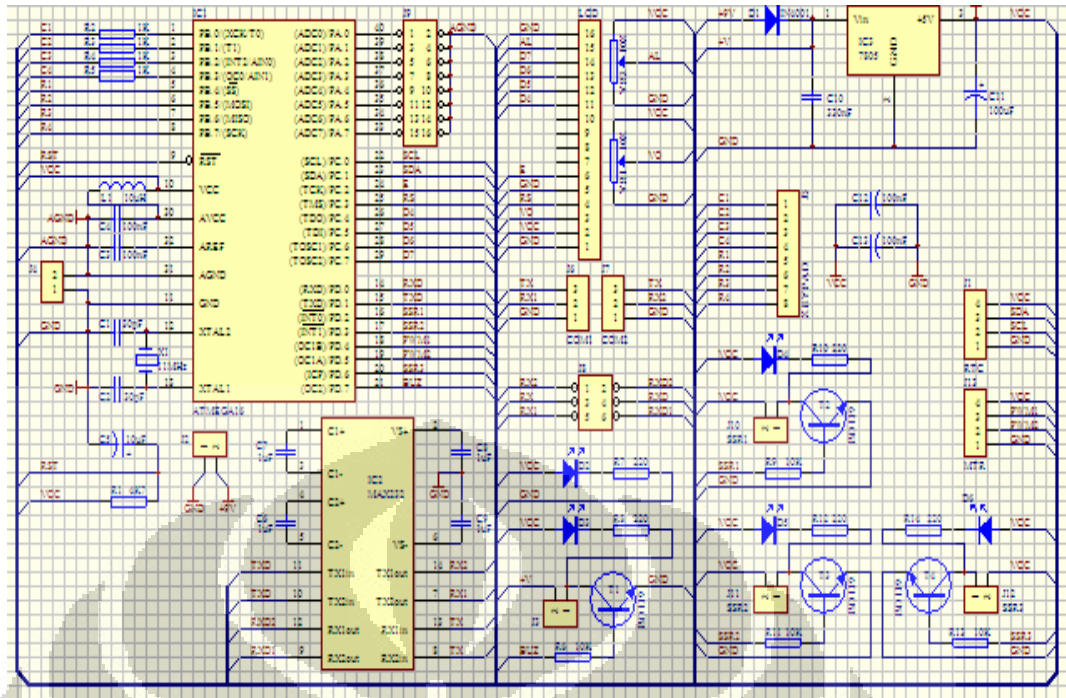
Gambar 3.3 Blok Diagram Pengendalian Motor

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 System*

Alat ini menggunakan *microcontroller* Atmega16 untuk pengendaliannya. Di dalam rangkaian *minimum system* menggunakan kristal 11MHz dan terdapat ISP AVR.



Gambar 3.4 Minimum System Microcontroller AVR

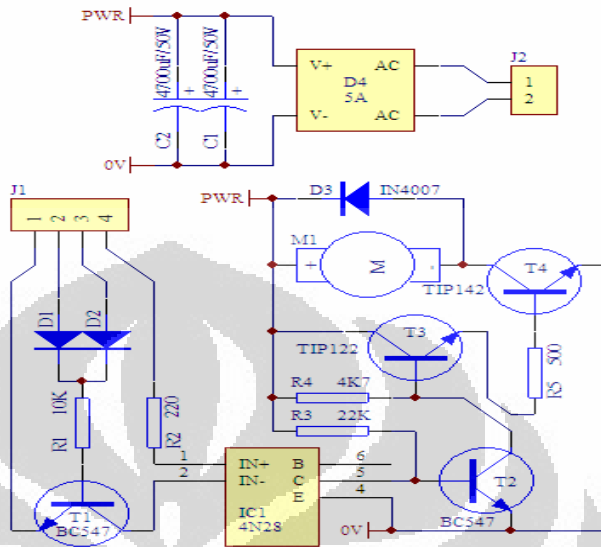
Rangkaian ini dibutuhkan agar dapat mengisi (me-download) program yang telah dibuat. Program tersebut meliputi program pengendalian suhu dan kecepatan serta pengiriman data komunikasi serial. Port yang digunakan pada *microcontroler* Atmega16 adalah PA untuk LCD, PC untuk keypad dan PD untuk komunikasi, motor dan *heater*.

3.2.2 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 pengendli 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. 3 merupakan rangkaian pengendali yang dapat mengendalikan kecepatan putaran sebuah motor DC.

Adapun cara kerja rangkaian motor driver ini adalah jika pada kaki 2 atau 3 pada J1 diberi logika 1 (atau diberi tegangan 5 Volt), maka BC 547 (sebagai saklar) akan aktif

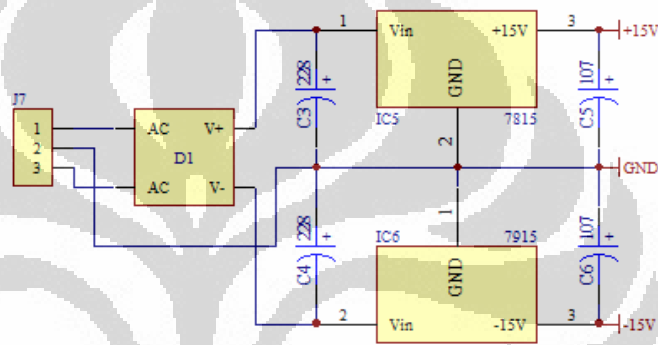
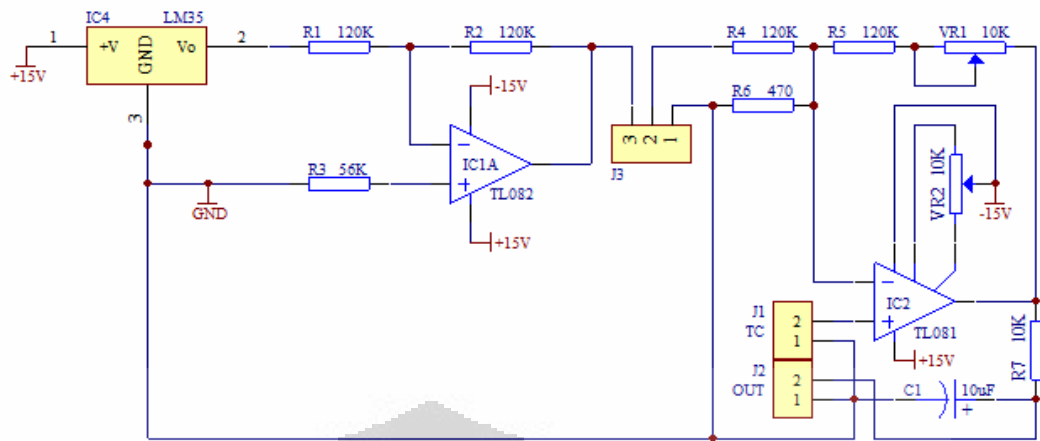
sehingga akan memberikan nilai pada M sesuai dengan inputan AC dari Dioda Bridge yaitu sebesar 24 Volt DC dan motor akan aktif.



Gambar 3.5 Rangkaian Driver Motor

3.2.3 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 280mV bila suhu ruangan adalah 28°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 (280mV ditambah 700mV).



Gambar 3.6 Rangkaian Cold Junction

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 dan juga pada labview. Setelah mendapatkan model dan mengetahui fungsi transfers dari data temperatur maka dengan menggunakan metode *Direct Synthesis* dibuat program Bascom AVR untuk mengendalikan sistem tersebut.

3.3.1 Proses pada Keypad

Dapat dijelaskan bahwa pada awal proses, program akan menginisialisai variabel-variabel yang digunakan pada program keypad tersebut, pada keadaan awal ini LCD akan menampilkan tulisan pada upperline "TEMPERATURE", dan pada lowerline "CONTROLLER V:2.0", lalu proses selanjutnya dengan memasukkan nilai parameter Temp, Time, dan Power Motor. Setelah memasukkan nilai-nilai *setpoint* tersebut, akan

disimpan setelah menekan tombol “#”. Ketika tombol “*” ditekan, maka seluruh proses kerja alat akan berjalan. Apa bila alat sedang bekerja, perubahan masukan hanya dapat dilakukan perubahan temperatur, Power Motor, dan Time. Saat proses berjalan, data dikirimkan ke Bascom Atmega16 untuk ditampilkan pada LCD. Sehingga perubahan data saat proses berjalan dapat diketahui. Dan apa bila tombol “#” ditekan maka keseluruhan proses akan berhenti. Motor dan heater dapat berhenti bekerja apa bila waktu habis dan atau tombol “#” ditekan. Apa bila tombol “*” di tekan, *Shaking Water Bath* dapat berlangsung dijalankan tanpa mengatur temperatur, Power Motor, dan Time, karena pengaturan pengendali sebelumnya sudah disimpan dimemori dan dapat langsung dipanggil..

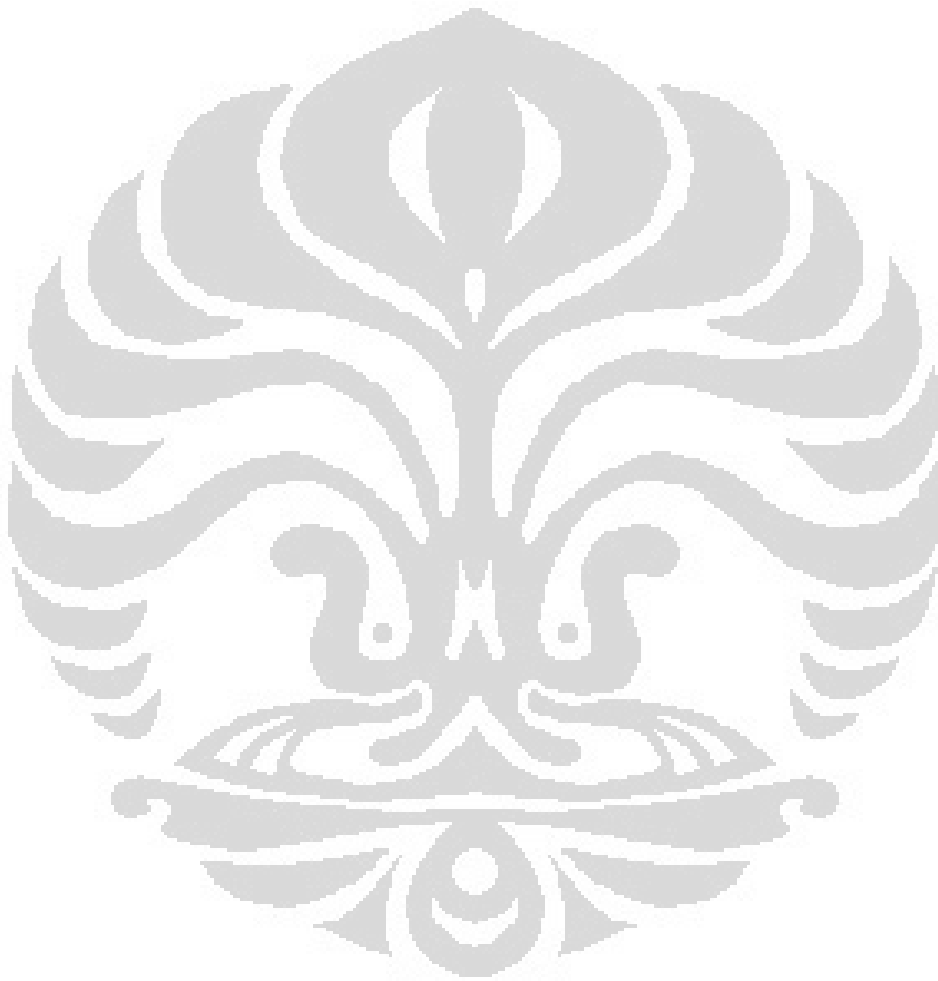
3.3.2 Pengambilan Data

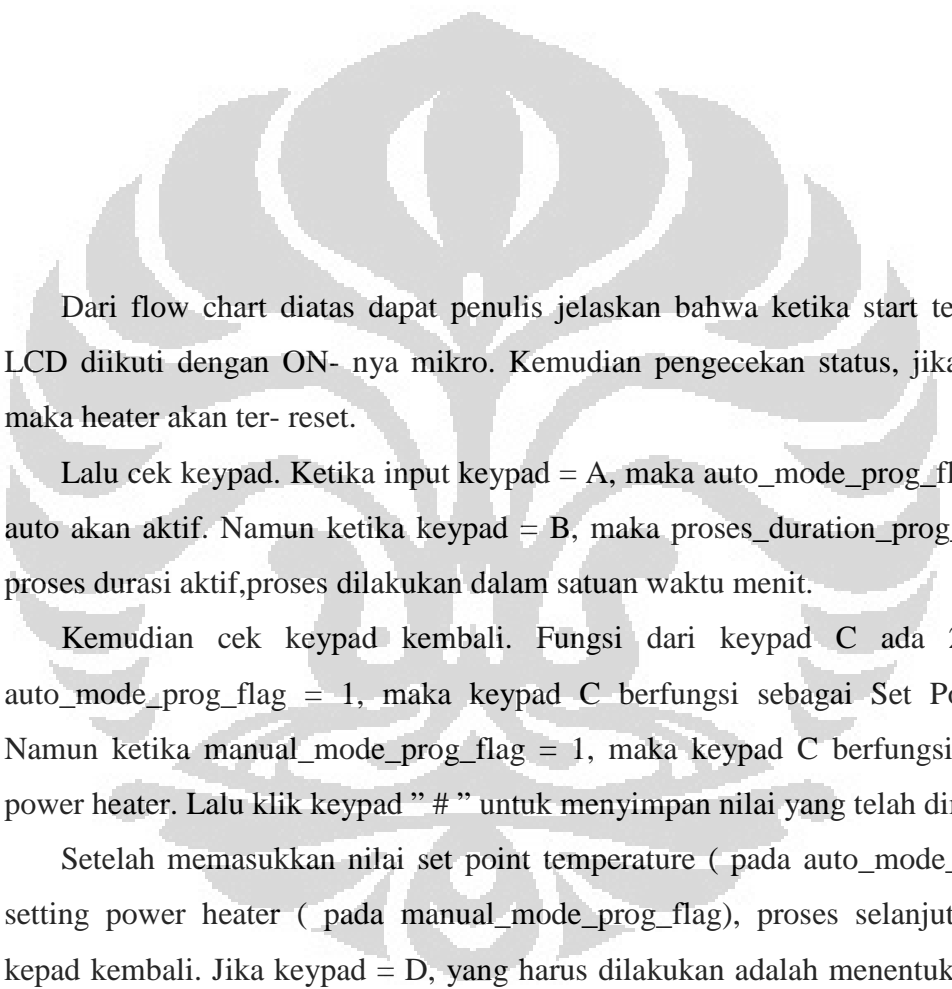
Ada tiga proses untuk mengambil data, yaitu untuk mengambil data temperatur, power motor dan waktu. Bila salah satu dari tombol perintah ditekan maka *lowerline* pada LCD akan berubah sesuai dengan tombol yang ditekan. Kemudian memasukkan nilai temperatur, power motor, dan waktu yang diinginkan dan sistem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

3.4 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. Apa bila 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 :





Dari flow chart diatas dapat penulis jelaskan bahwa ketika start terjadi inisialisasi LCD diikuti dengan ON- nya mikro. Kemudian pengecekan status, jika `auto_flag = 1`, maka heater akan ter- reset.

Lalu cek keypad. Ketika input keypad = A, maka `auto_mode_prog_flag = 1`, program auto akan aktif. Namun ketika keypad = B, maka `proses_duration_prog_flag = 1`, maka proses durasi aktif, proses dilakukan dalam satuan waktu menit.

Kemudian cek keypad kembali. Fungsi dari keypad C ada 2, yaitu ketika `auto_mode_prog_flag = 1`, maka keypad C berfungsi sebagai Set Point temperatur. Namun ketika `manual_mode_prog_flag = 1`, maka keypad C berfungsi sebagai setting power heater. Lalu klik keypad " # " untuk menyimpan nilai yang telah dimasukkan.

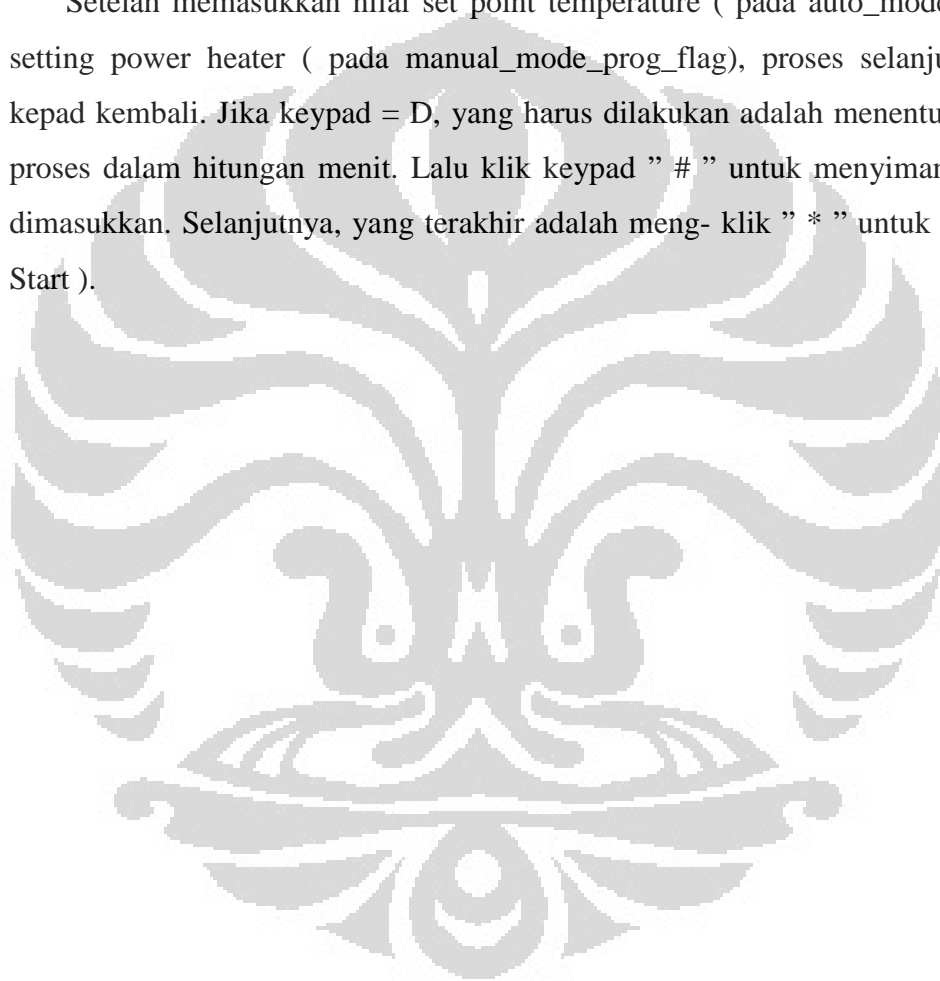
Setelah memasukkan nilai set point temperature (pada `auto_mode_prog_flag`) atau setting power heater (pada `manual_mode_prog_flag`), proses selanjutnya adalah cek keypad kembali. Jika keypad = D, yang harus dilakukan adalah menentukan power motor untuk menentukan berapa cepatnya motor berputar. Lalu klik keypad " # " untuk menyimpan nilai yang telah dimasukkan. Selanjutnya, yang terakhir adalah meng- klik " * " untuk memulai proses (Start).

Dari flow chart diatas dapat penulis jelaskan bahwa ketika start terjadi inisialisasi LCD diikuti dengan ON- nya mikro. Kemudian pengecekan status, jika `auto_flag = 1`, maka heater akan ter- reset.

Lalu cek keypad. Ketika input keypad = A, maka `auto_mode_prog_flag = 1`, program auto akan aktif. Namun ketika keypad = B, maka `manual_mode_prog_flag = 1`, maka program manual yang akan aktif.

Kemudian cek keypad kembali. Fungsi dari keypad C ada 2, yaitu ketika `auto_mode_prog_flag = 1`, maka keypad C berfungsi sebagai Set Point temperatur. Namun ketika `manual_mode_prog_flag = 1`, maka keypad C berfungsi sebagai setting power heater. Lalu klik keypad " # " untuk menyimpan nilai yang telah dimasukkan.

Setelah memasukkan nilai set point temperature (pada `auto_mode_prog_flag`) atau setting power heater (pada `manual_mode_prog_flag`), proses selanjutnya adalah cek keypad kembali. Jika keypad = D, yang harus dilakukan adalah menentukan durasi waktu proses dalam hitungan menit. Lalu klik keypad " # " untuk menyimpan nilai yang telah dimasukkan. Selanjutnya, yang terakhir adalah meng- klik " * " untuk memulai proses (Start).



BAB 4

PENGUJIAN SISTEM DAN ANALISA

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 *Thermocouple*
- Pengujian Sistem Kendali dengan Metode *Ziegler Nichols*

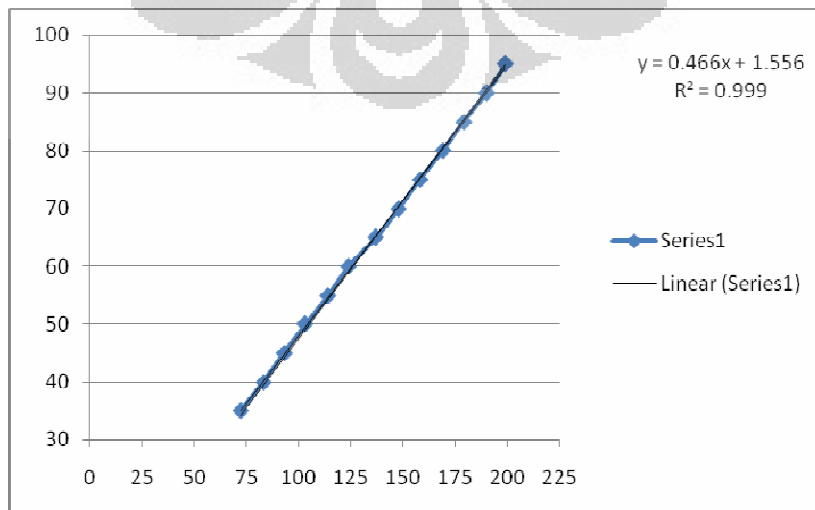
4.1 Pengujian *Thermocouple*

Pada pengujian *Thermocouple* ini bertujuan untuk mengetahui seberapa besar °C yang kemudian juga untuk mengetahui nilai ADC yang diukur oleh *thermocouple* bila pada heater diberikan daya mencapai 100%. Pengujian ini dilakukan menggunakan air. Pengujian ini dilakukan dari 95°C hingga 35°C setiap perubahan turun 5°C. Pengukuran temperaturnya diukur dan dibandingkan oleh termometer analog dan temperatur pada *thermocouple* menggunakan cara manual yakni dengan menampilkan langsung nilai bit ADC pada LCD untuk pengambilan data ADC. Karena proses pengambilan data secara manual maka pengambilan data dilakukan sebanyak 3 kali. Dilakukan 3 kali untuk mendapatkan hasil data ADC yang sesuai. Jadi ketika sudah selesai melakukan pengambilan data sebanyak 3 kali, hasil ketiga data tersebut kita rata-rata kan. Dan dapat dilihat hasil data yang sudah di rata-ratakan adalah antara ADC1,ADC2, dan ADC3 hasilnya tidak jauh berbeda. Jadi dapat disimpulkan ADC yang didapatkan adalah stabil. Setelah melakukan percobaan tersebut maka hasil yang didapatkan adalah sebagai berikut :

Tabel 2.3 Data ADC

Data ADC1		Data ADC2		Data ADC3		Data ADC rata2	
ADC	Temp	ADC	Temp	ADC	Temp	ADC	Temp
199	95	199	95	198	95	199	95
190	90	191	90	189	90	190	90
180	85	179	85	179	85	179	85
169	80	169	80	168	80	169	80
158	75	159	75	158	75	158	75
148	70	148	70	147	70	148	70
137	65	137	65	137	65	137	65
124	60	124	60	125	60	124	60
114	55	114	55	114	55	114	55
103	50	104	50	103	50	103	50
93	45	93	45	93	45	93	45
83	40	84	40	83	40	83	40
72	35	73	35	72	35	72	35

Berdasarkan tabel 2.3 diatas maka bentuk dari grafik ADCnya adalah sebagai berikut:



Gambar 4.1 Grafik Persamaan ADC

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

$$y = 0,466 x - 1,556 \dots \dots \dots (4.1)$$

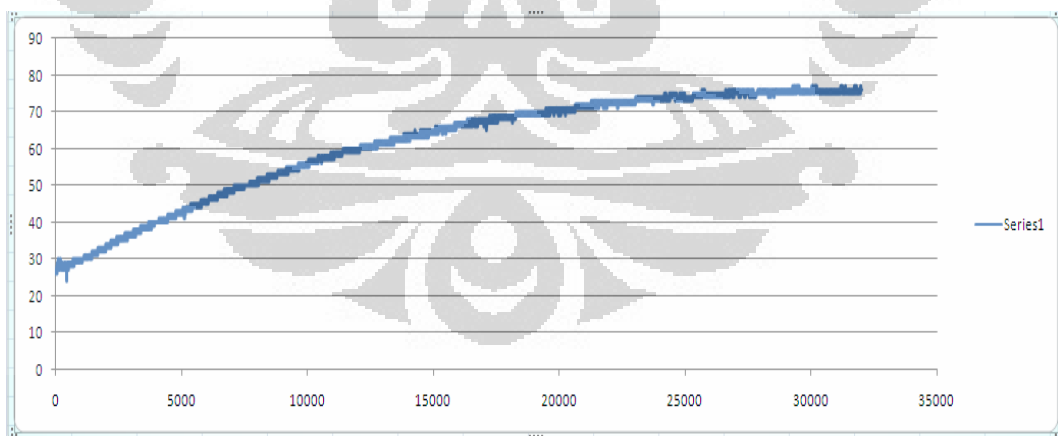
$$R^2 = 0,999$$

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

$$x = \frac{y + 1,556}{0,466} \dots \dots \dots (4.2)$$

Dimana x adalah nilai temperatur yang terukur pada termometer digital dan y adalah nilai bit dari ADC. Dari persamaan garis didapatkan nilai $R^2 = 0,999$, artinya sensor temperatur yang digunakan dalam pengukuran temperaturnya dapat dikatakan baik.

Setelah didapatkan rata-rata temperatur maka diuji kembali dengan mencoba memanaskan heater dengan daya 20%. Dengan demikian didapatkan grafik:



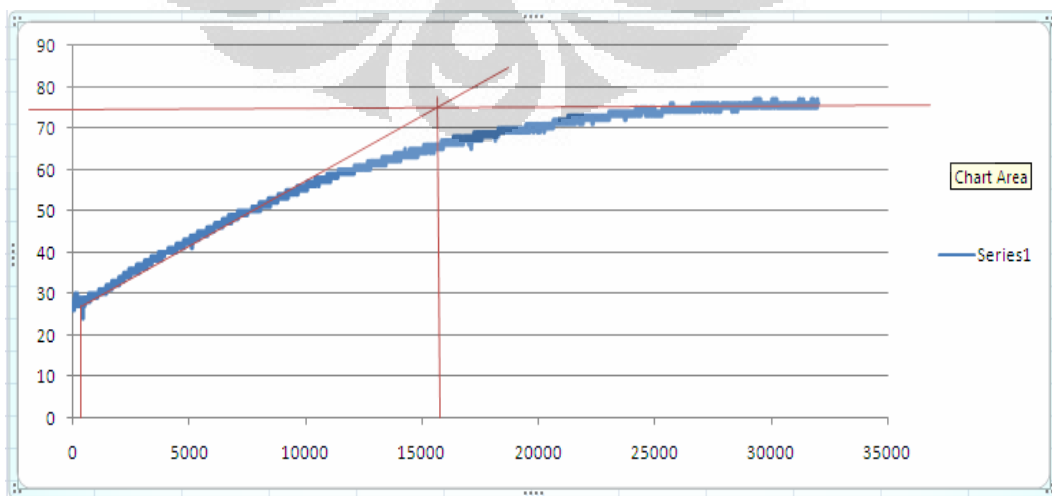
Gambar 4.2 Respon waktu proses perubahan temperatur

4.2 Pengujian Sistem Pengendalian Dengan Metode Ziegler Nichols

Pengujian sistem pengendali dengan metode *Ziegler Nichols* berfungsi mencari fungsi transfer dengan mencoba sistem tersebut. Dari Grafik daya 20% pada pengujian termokopel merupakan langkah awal yang dilakukan untuk menentukan nilai matematik dari respon sistem, untuk mendapatkan fungsi transfernya harus mencari nilai L dan nilai T. Untuk mendapatkan nilai L, 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 grafik. Pada titik temu tersebut ditarik garis vertikal hingga diketahui posisi garis vertikal tersebut. Nilai L adalah nilai yang ditunjukkan garis tersebut. Sedangkan untuk Mendapatkan nilai T harus mendapatkan sebuah titik terlebih dahulu yakni dengan perhitungan seperti dibawah ini :

$$\text{Titik} = PV1 + (0,632 \times \Delta PV) \dots\dots\dots(4.3)$$

Dimana PV1 adalah batas minimum temperatur pada Gambar 4.2 dan ΔPV adalah nilai selisih antara batas maksimum temperatur dengan batas minimum temperatur pada Grafik 4.2. Dengan melakukan perhitungan seperti persamaan (4.3) maka didapatkan nilai titik tersebut yakni 58. 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 T adalah selisih dari nilai yang ditunjukkan garis tersebut dengan nilai L. Untuk mencari nilai L dan nilai T juga dapat melihat Gambar 4.3.



Gambar 4.3 Mencari Nilai L dan Nilai T

Bila dilihat dari gambar diatas, dapat diketahui fungsi transfer dari system lalu dengan menggunakan metode *zigher nichols* dapat diperoleh nilai Kp, Ti dan Td. Sebelum itu, dari gambar diatas diperoleh nilai L= 51 dan nilai T = 15296 – 51 = 15145. Sedangkan nilai $\Delta MV = 20 - 0 = 20\%$ dan $\Delta PV = 76 - 26 = 50$.

Dengan demikian persamaan yang didapat :

$$MV = Kp \left(E + \frac{1}{Ti} \int E dt + Td \frac{dE}{dt} \right) \dots\dots\dots(4.4)$$

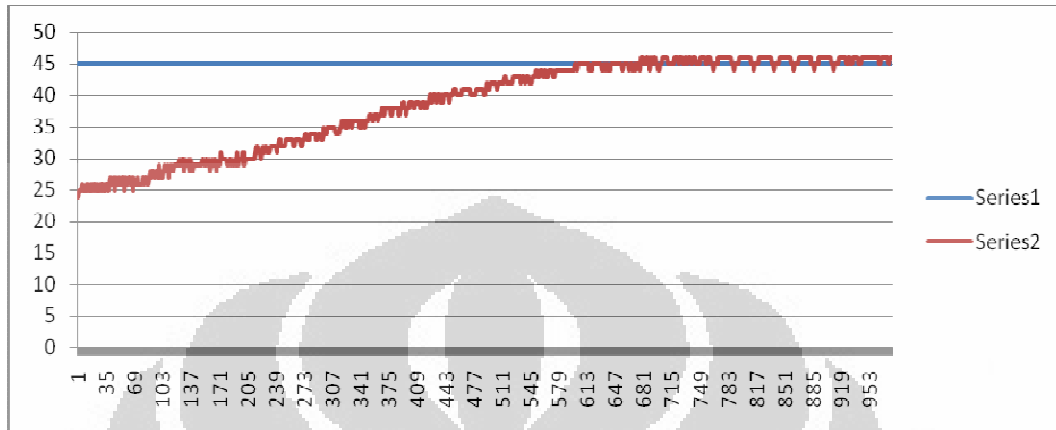
Dari persamaan diatas dapat mencari nilai Kp, Ti, dan Td dengan menggunakan persamaan-persamaan dibawah ini :

$$\begin{aligned} \Delta PV &= PV1 - PV0 & Ti &= 2. L \\ &= 76^\circ - 26^\circ & &= 2. 51 \\ &= 50 & &= 102 \\ L &= 51 & Td &= 0.5 * L \\ N &= \Delta PV / T & &= 0.5 * 51 \\ &= 50 / (15196 - 51) & &= 25,5 \\ &= 0.003 \\ Kp &= (1.2 * \Delta CV) / (N * L) \\ &= (1.2 * 26) / (0.003 * 51) \\ &= 204 \end{aligned}$$

Setelah itu, nilai dari persamaan-persamaan diatas yang telah didapatkan kemudian dimasukkan kedalam program pengendalian. Kemudian didapatkan data seperti yang tertera pada Tabel 3.8 yang terdapat pada lampiran. Berdasarkan dari data tersebut dapat diketahui bahwa nilai temperatur yang didapatkan mengalami sedikit *over shoot*.

4.3 Hasil Pengujian Sistem Pengendali dengan SP= 45° c

Hasil pengujian sistem pengendali dengan SP=45°c sebagai berikut :



Gambar 4.4 Grafik Pengendalian temperatur dengan SP=45°c

Berdasarkan dari hasil grafik diatas dapat dilihat bahwa dengan set poin 45° dengan waktu kurang lebih 90 menit, temperatur sudah mendekati titik stabil atau sesuai dengan set poin. Hanya terjadi sedikit over shoot sekitar 2° c.

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah menyelesaikan perancangan peralatan serta pengambilan data, maka penulis dapat menarik kesimpulan bahwa :

- 1) Alat ini berjalan cukup baik dalam proses pemanasan dan pengadukkan dengan mengatur temperatur dan kecepatan keranjangnya sesuai.
- 2) Diperoleh nilai $R^2 = 0.999$ yang berarti bahwa sensor suhu bekerja dengan baik.
- 3) Alat ini menggunakan termokopel tipe K.
- 4) Alat ini menggunakan *Time Sampling* pengendalian yang digunakan sebesar 1 s.
- 5) Dengan *ziegler nichols method* didapat persamaan pengendali berupa pengendali PID dengan nilai $K_p = 204$; $T_i = 102$ dan $T_d = 25,5$.
- 6) Bila diberi daya 20% suhu akan stabil saat 76°C .
- 7) Semakin besar tegangan yang diberikan pada motor maka semakin besar pula nilai frekuensi dan kecepatan rpmnya.

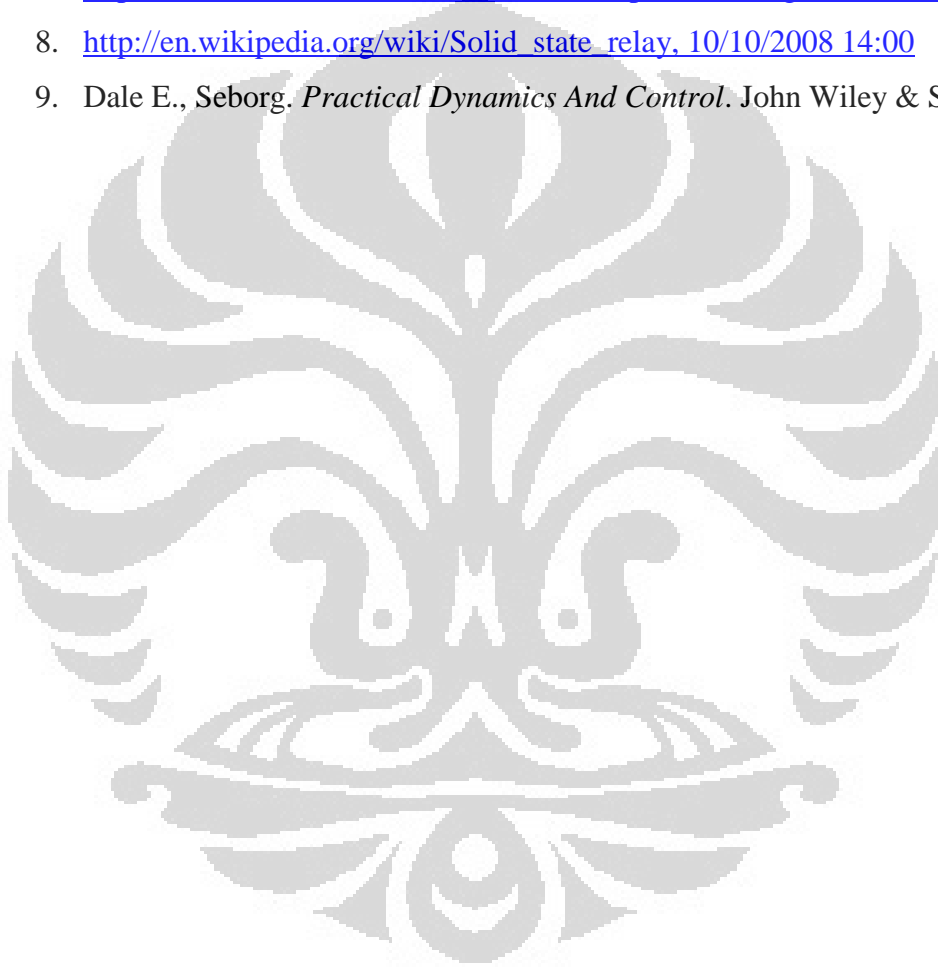
5.2 Saran

1. Heater ini mempunyai temperatur maksimal hanya sampai 1000°C , jadi sebaiknya kontrol untuk heater jangan sampai melebihi temperatur tersebut.
2. Sebaiknya dalam pengambilan data agar didapat nilai hasil pengukuran yang akurat kita menggunakan dua alat ukur (termometer digital dan termometer analog).
3. Dilakukan percobaan berulang-ulang agar mendapatkan hasil yang akurat untuk system pengendalian.

DAFTAR ACUAN

1. <http://www.ent.ohio.edu/~bobw/PDF/IntroRob.pdf>, 10/8/2008 16:25

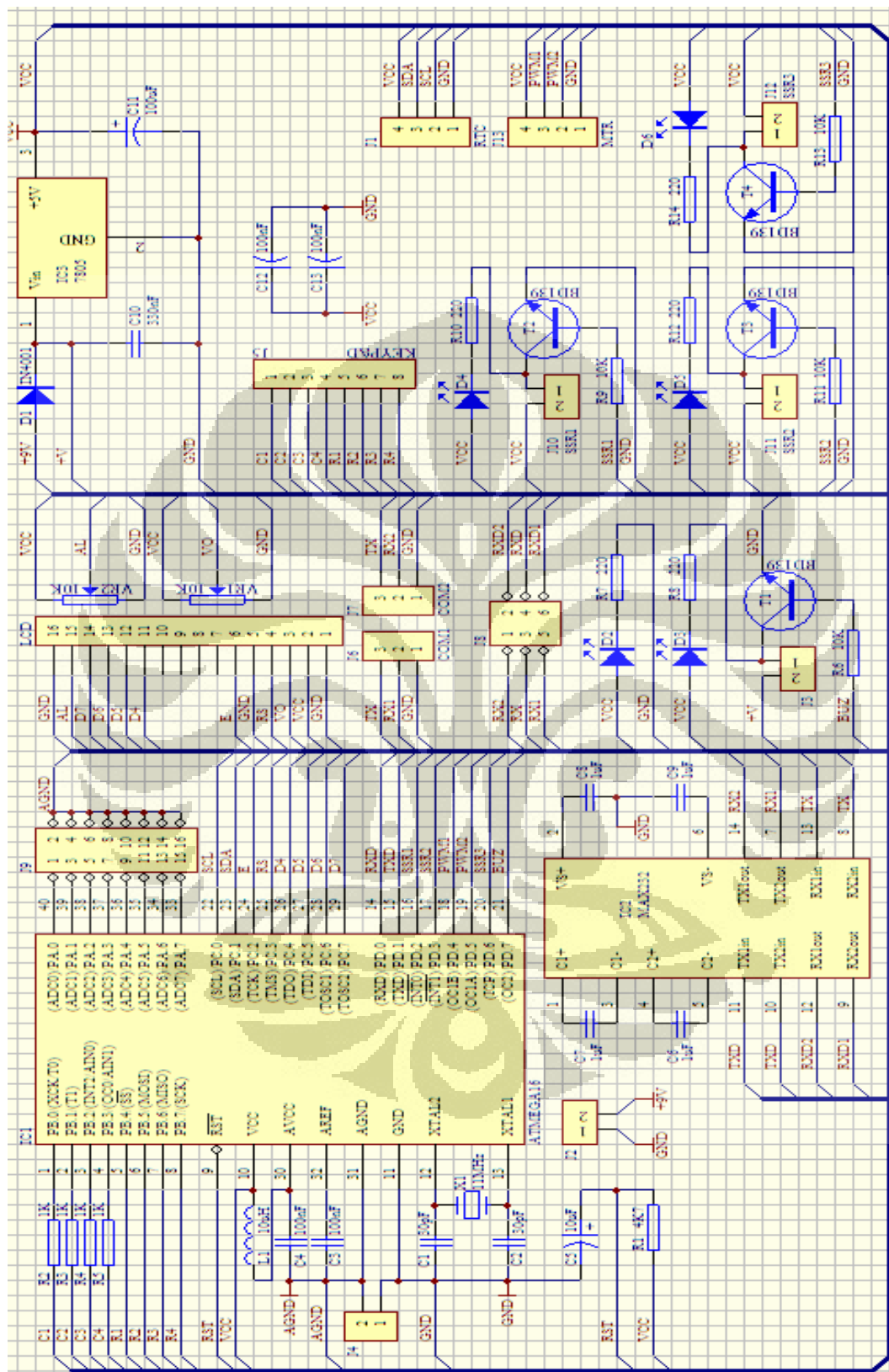
2. http://www.asia.ru/Catalog/?page=28category_id=935, 10/8/2008 16:27 (Gambar heater hot plate)
3. <http://www.omega.com/referenceTemperatur/TheThermocouple/pdf>, 10/8/2008 16:10
4. <http://en.wikipedia.org/wiki/Thermocouple>, 10/8/2008 16:15
5. Kilian. *Modern Control Technology : Components and Systems*. Delmar, Inc.
6. Clayton, George. *Operational Amplifier*. Erlangga, 2002.
7. <http://www.toko-elektronika.com/tutorial /pcinterfacing.htm>, 10/10/2008 11:30
8. http://en.wikipedia.org/wiki/Solid_state_relay, 10/10/2008 14:00
9. Dale E., Seborg. *Practical Dynamics And Control*. John Wiley & Sons Inc, 1989.



The image features a large, light gray watermark of the Universitas Indonesia logo, which is a stylized, symmetrical floral or geometric design. Overlaid on this watermark is the word "LAMPIRAN" in a bold, black, serif font, centered horizontally.

LAMPIRAN

Universitas Indonesia





PROGRAM PENGAMBILAN DATA

\$regfile = "m16DEF.dat"

\$crystal = 11059200

\$baud = 19200

On Ovf0 Time_base

On Urxc Serial_in

Enable Interrupts

Enable Ovf0

Enable Urxc

Config Timer0 = Timer , Prescale = 8

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

Pwm = Clear Down , Prescale = 8

Config Lcd = 16 * 2

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

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

Config Kbd = Portb.debounce = 40

Config Portd.2 = Output

Config Portd.3 = Output

Config Portd.6 = Output

Config Portd.7 = Output

Buzzer Alias Portd.7

Heater1 Alias Portd.2

Heater2 Alias Portd.3

Heater3 Alias Portd.5

Dim Auto_flag As Bit

Dim Auto_prog_flag As Bit

Dim Manual_flag As Bit

Dim Manual_prog_flag As Bit

Dim Start_flag As Bit

Dim Stop_flag As Bit

Dim Data_flag As Bit

Dim Balik As Bit

Dim Ulang As Bit

Dim Periode_1s_flag As Bit

Dim Get_data_flag As Bit

Dim Periode_1s As Byte

Dim Periode_10ms As Byte

Dim Keypad As Byte

Dim Mv_var As Byte
Dim Power_heater As Byte
Dim Iterasi As Byte
Dim Nilai_keypad As Byte
Dim Temperature As Word
Dim Nilai_adc As Word
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 Temp_lcd As Integer
Dim Waktu As Integer
Dim Batas_atas As Single
Dim Batas_bawah As Single
Dim Command As String * 9
Dim Nilai_str As String * 7
Dim Data_ser_i As String * 1

Auto_prog_flag = 1

Manual_prog_flag = 0

Main_prog:

Start_flag = 0

Stop_flag = 0

Cursor Off

Cls

Locate 1 , 1

Lcd " TEMPERATURE "

Locate 2 , 1

Lcd "CONTROLLER V:2.0"

Balik = 1

Reset Heater1

Reset Heater2

Reset Heater3

Do

Loop Until Balik = 0

If Start_flag = 1 Then Goto Start_prog

Cls

Locate 1 , 1

Lcd "UNKNOWN TRIGGER "

Locate 2 , 1

Lcd " IN CONTROLLER "

Wait 2

Goto Main_prog

```
Time_base:
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
    Reset Heater3
  Else
    Set Heater1
    Set Heater2
    Set Heater3
    Power_heater = Power_heater - 1
  End If
  If Periode_1s = 100 Then
    Periode_1s_flag = 1
    Periode_1s = 0
    Power_heater = Mv_var
  End If
End If
Return
```

```
Serial_in:
Disable Interrupts
Data_seri = Inkey()
If Data_seri = "*" Then
  Balik = 0
  Data_seri = Waitkey()
```

```
If Data_seri = "A" Then
    Auto_prog_flag = 1
    Manual_prog_flag = 0
    Start_flag = 1
    Stop_flag = 0
    Data_seri = Waitkey()
    Nilai_str = ""
    Ulang = 1
    Do
        Data_seri = Waitkey()
        If Data_seri = "#" Then
            Ulang = 0
            Temperature = Val(nilai_str)
        Else
            Nilai_str = Nilai_str + Data_seri
        End If
    Loop Until Ulang = 0
End If
If Data_seri = "M" Then
    Auto_prog_flag = 0
    Manual_prog_flag = 1
    Start_flag = 1
    Stop_flag = 0
    Data_seri = Waitkey()
    Nilai_str = ""
    Ulang = 1
    Do
        Data_seri = Waitkey()
        If Data_seri = "#" Then
            Ulang = 0
```



```

        Mv_var = Val(nilai_str)
    Else
        Nilai_str = Nilai_str + Data_seri
    End If
    Loop Until Ulang = 0
End If
If Data_seri = "G" Then Get_data_flag = 1
If Data_seri = "S" Then
    Start_flag = 0
    Stop_flag = 1
    Ulang = 1
    Do
        Data_seri = Waitkey()
        If Data_seri = "#" Then Ulang = 0
    Loop Until Ulang = 0
End If
End If
Enable Interrupts
Return

```

```

Start_prog:
If Auto_prog_flag = 1 Then
    Start Adc
    Start Timer0
    Error = 0
    Sigma_error_kp = 0
    Temp_lcd = 0
    Kp = 0.5
    Ti = 2

```

```

Td = 0.2
Waktu = 0
Cls
Locate 1 , 1
Lcd " AUTOMATIC MODE "
Locate 2 , 1
Lcd "SP=" ; Temperature ; Chr(223) ; "C"
Locate 2 , 10
Lcd "T=" ; Temp_lcd ; Chr(223) ; "C"
Do
  If Periode_1s_flag = 1 Then
    Periode_1s_flag = 0
    Error_old = Error
    Nilai_adc = Getadc(0)
    Nilai_temp = Nilai_adc * 0.476
    Nilai_temp = Round(nilai_temp)
    Temp_lcd = Nilai_temp
    Sp = Temperature
    Pv = Nilai_temp
    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
    Mv_i = Sigma_error_kp / Ti
    Mv_d = Kp * Delta_error
    Mv_d = Mv_d * Td
    Mv = Mv_p + Mv_i

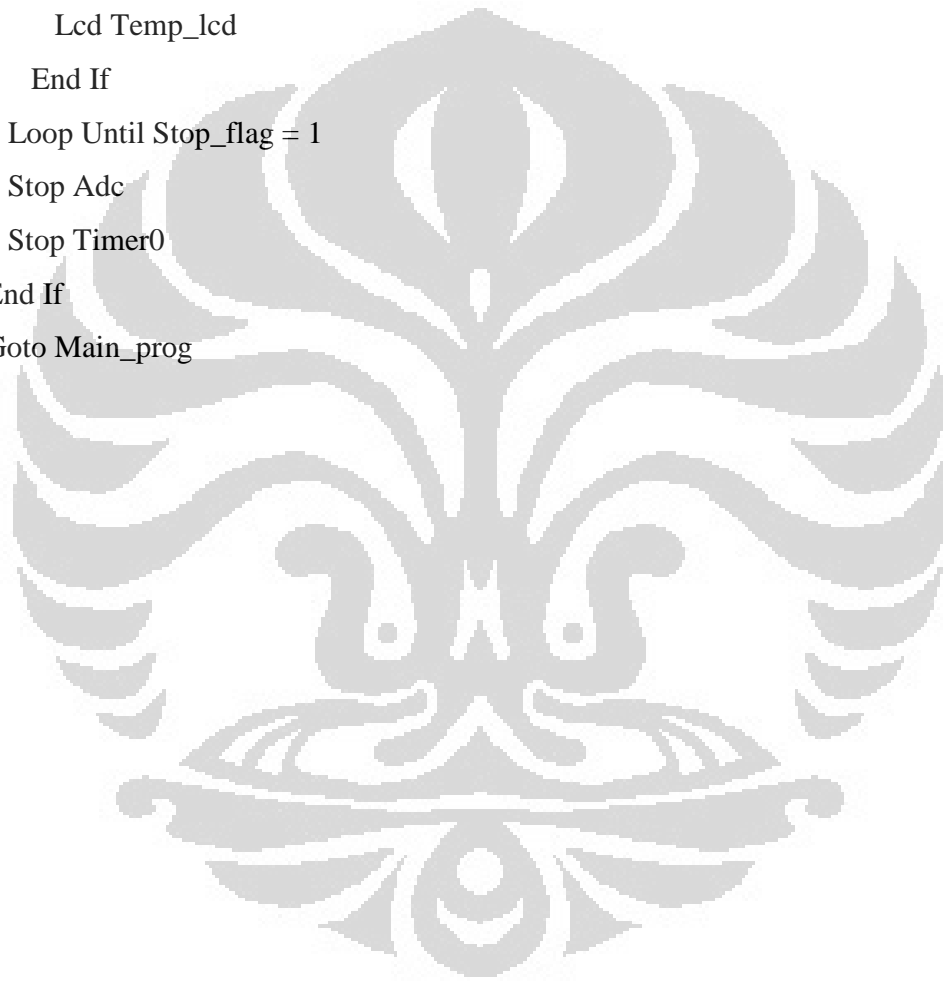
```

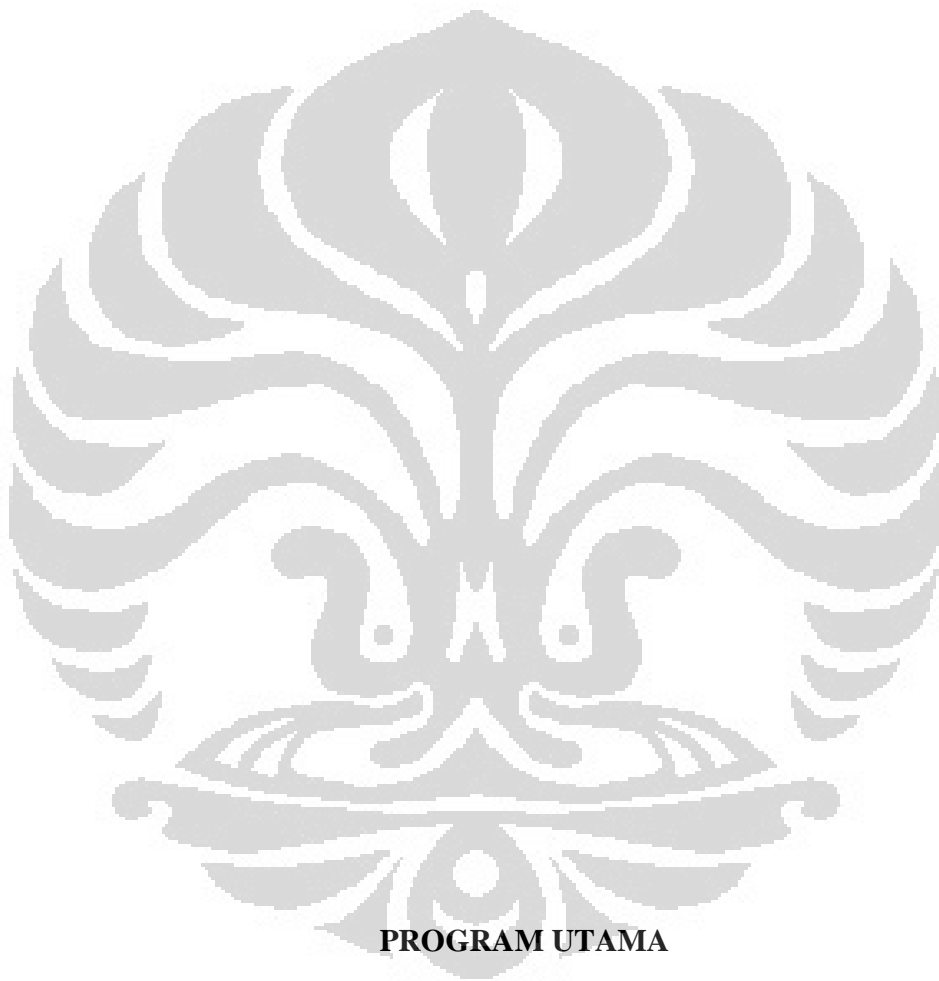
```

Mv = Mv + Mv_d
If Mv > 100 Then Mv = 100
If Mv < 0 Then Mv = 0
Mv_var = Mv
Locate 2 , 12
Lcd "  "
Locate 2 , 12
Lcd Temp_lcd ; Chr(223) ; "C"
End If
Loop Until Stop_flag = 1
Stop Adc
Stop Timer0
End If
If Manual_prog_flag = 1 Then
Start Adc
Start Timer0
Cls
Locate 1 , 1
Lcd " MANUAL MODE "
Locate 2 , 1
Lcd "MV=" ; Mv_var ; "%"
Locate 2 , 10
Lcd "T=" ; Temp_lcd
Locate 2 , 15
Lcd Chr(223) ; "C"
Do
If Get_data_flag = 1 Then
Get_data_flag = 0
Nilai_adc = Getadc(0)
Nilai_temp = Nilai_adc * 0.476

```

```
Nilai_temp = Round(nilai_temp)
Temp_lcd = Nilai_temp
Print "*" ; Temp_lcd ; "#"
Locate 2 , 12
Lcd " "
Locate 2 , 12
Lcd Temp_lcd
End If
Loop Until Stop_flag = 1
Stop Adc
Stop Timer0
End If
Goto Main_prog
```





PROGRAM UTAMA

```
$regfile = "m16DEF.dat"
```

```
$crystal = 11059200
```

```
$baud = 19200
```

```
On Ovf0 Time_base
```

Universitas Indonesia

On Urxc Serial_in

Enable Interrupts

Enable Ovf0

Enable Urxc

Config Timer0 = Timer , Prescale = 8

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

Pwm = Clear Down , Prescale = 256

Config Lcd = 16 * 2

Config Lcdpin = Pin , Db4 = Portc.4 , Db5 = Portc.5 , Db6 = Portc.6 , Db7 = Portc.7 ,

E = Portc.2 , Rs = Portc.3

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

Config Kbd = Portb.debounce = 40

Config Portd.2 = Output

Config Portd.3 = Output

Config Portd.6 = Output

Config Portd.7 = Output

Heater1 Alias Portd.2

Heater2 Alias Portd.3

Heater3 Alias Portd.6

Buzzer Alias Portd.7

Dim Kirim_flag As Bit

Dim Mode_prog_flag As Bit
Dim Time_prog_flag As Bit
Dim Data_temp_prog_flag As Bit
Dim Start_prog_flag As Bit
Dim Data_motor_prog_flag As Bit
Dim Auto_flag As Bit
Dim Start_flag As Bit
Dim Stop_flag As Bit
Dim Data_flag As Bit
Dim Balik As Bit
Dim Ulang As Bit
Dim Periode_1s_flag As Bit
Dim Periode_1s As Byte
Dim Periode_10ms As Byte
Dim Keypad As Byte
Dim Mv_var As Byte
Dim Power_heater As Byte
Dim Power_motor As Byte
Dim Power_blower As Byte
Dim Iterasi As Byte
Dim Nilai_keypad As Byte
Dim Temperature As Word
Dim Periode As Word
Dim Nilai_adc As Word
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 Temp_lcd As Integer
Dim Waktu As Long
Dim Data_seri As String * 1
Dim Command As String * 1
Dim Nilai_str As String * 3

Seeprom

Data_controller:

Data 30 , 30 , 30 , 30

\$data

Auto_flag = 1

Main_prog:

Mode_prog_flag = 0

Time_prog_flag = 0

Data_temp_prog_flag = 0

Data_motor_prog_flag = 0

Start_flag = 0

Stop_flag = 0


```
Cursor Off
Cls
Locate 1 , 1
Lcd " TEMPERATURE  "
Locate 2 , 1
Lcd "CONTROLLER V:2.0"
Balik = 1
Reset Heater1
Reset Heater2
Reset Heater3
Do
  Keypad = Getkbd()
  If Keypad < 16 Then
    Select Case Keypad
      Case 12
        Mode_prog_flag = 1
      Case 8
        Time_prog_flag = 1
      Case 4
        Data_temp_prog_flag = 1
      Case 3
        Start_prog_flag = 1
      Case 0
        Data_motor_prog_flag = 1
    End Select
  Ulang = 1
Do
  Keypad = Getkbd()
  If Keypad = 16 Then Ulang = 0
Loop Until Ulang = 0
```

```
Balik = 0
End If
Loop Until Balik = 0
If Data_temp_prog_flag = 1 Then Goto Data_temp_prog
If Data_motor_prog_flag = 1 Then Goto Data_motor_prog
If Mode_prog_flag = 1 Then Goto Mode_prog
If Time_prog_flag = 1 Then Goto Time_prog
If Start_prog_flag = 1 Then Goto Start_prog
Goto Main_prog

Mode_prog:
If Auto_flag = 1 Then
    Auto_flag = 0
    Cls
    Locate 1 , 1
    Lcd " CONTROLLER "
    Locate 2 , 1
    Lcd " ON MANUAL MODE "
    Wait 2
Else
    Auto_flag = 1
    Cls
    Locate 1 , 1
    Lcd " CONTROLLER ON "
    Locate 2 , 1
    Lcd " AUTOMATIC MODE "
    Wait 2
End If
Goto Main_prog
```

```

Data_temp_prog:
If Auto_flag = 1 Then
  Readeeprom Temperature , Data_controller
  Ulang = 1
  Iterasi = 0
  Cls
  Locate 1 , 1
  Lcd " AUTOMATIC MODE "
  Locate 2 , 1
  Lcd "TEMP = " ; Temperature ; " " ; Chr(223) ; "C"
  Do
    Keypad = Getkbd()
    If Keypad < 16 Then
      Set Buzzer
      Select Case Keypad
        Case 2
          Nilai_keypad = 0
        Case 15
          Nilai_keypad = 1
        Case 14
          Nilai_keypad = 2
        Case 13
          Nilai_keypad = 3
        Case 11
          Nilai_keypad = 4
        Case 10
          Nilai_keypad = 5
        Case 9

```

```

    Nilai_keypad = 6
Case 7
    Nilai_keypad = 7
Case 6
    Nilai_keypad = 8
Case 5
    Nilai_keypad = 9
Case 1
    Nilai_keypad = 10
End Select
Waitms 100
Reset Buzzer
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
        Temperature = Nilai_keypad
    Else
        Temperature = 10 * Temperature
        Temperature = Temperature + Nilai_keypad
    End If
End If
If Temperature > 250 Then Temperature = 250
Locate 2 , 8
Lcd "      "
Locate 2 , 8

```

```

    Lcd Temperature ; " " ; Chr(223) ; "C"
    If Nilai_keypad = 10 Then
        Ulang = 0
        Writeeprom Temperature , Data_controller
    End If
End If
Loop Until Ulang = 0
Else
    Readeeprom Temperature , Data_controller
    Readeeprom Mv_var
    Ulang = 1
    Iterasi = 0
    Cls
    Locate 1 , 1
    Lcd " MANUAL MODE "
    Locate 2 , 1
    Lcd "PWR HTR = " ; Mv_var ; " %"
    Do
        Keypad = Getkbd()
        If Keypad < 16 Then
            Set Buzzer
            Select Case Keypad
                Case 2
                    Nilai_keypad = 0
                Case 15
                    Nilai_keypad = 1
                Case 14
                    Nilai_keypad = 2
                Case 13
                    Nilai_keypad = 3
            End Select
        End If
    Loop

```

```
Case 11
    Nilai_keypad = 4
Case 10
    Nilai_keypad = 5
Case 9
    Nilai_keypad = 6
Case 7
    Nilai_keypad = 7
Case 6
    Nilai_keypad = 8
Case 5
    Nilai_keypad = 9
Case 1
    Nilai_keypad = 10
End Select
Waitms 100
Reset Buzzer
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
        Mv_var = Nilai_keypad
    Else
        Mv_var = 10 * Mv_var
        Mv_var = Mv_var + Nilai_keypad
    End If
End If
```

```

    If Mv_var > 100 Then Mv_var = 100
    Locate 2 , 11
    Lcd "    "
    Locate 2 , 11
    Lcd Mv_var ; " %"
End If
If Nilai_keypad = 10 Then
    Ulang = 0
    Writeeprom Temperature , Data_controller
    Writeeprom Mv_var
End If
End If
Loop Until Ulang = 0
End If
Goto Main_prog

Data_motor_prog:
Readeeprom Temperature , Data_controller
Readeeprom Mv_var
Readeeprom Power_motor
Cls
Locate 1 , 1
Lcd " SHAKIMG WATER BATH "
Locate 2 , 1
Lcd "PWR MTR = " ; Power_motor ; " %"
Ulang = 1
Iterasi = 0
Do
    Keypad = Getkbd()

```

```
If Keypad < 16 Then
  Set Buzzer
  Select Case Keypad
    Case 2
      Nilai_keypad = 0
    Case 15
      Nilai_keypad = 1
    Case 14
      Nilai_keypad = 2
    Case 13
      Nilai_keypad = 3
    Case 11
      Nilai_keypad = 4
    Case 10
      Nilai_keypad = 5
    Case 9
      Nilai_keypad = 6
    Case 7
      Nilai_keypad = 7
    Case 6
      Nilai_keypad = 8
    Case 5
      Nilai_keypad = 9
    Case 1
      Nilai_keypad = 10
  End Select
  Waitms 100
  Reset Buzzer
  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
        Power_motor = Nilai_keypad
    Else
        Power_motor = 10 * Power_motor
        Power_motor = Power_motor + Nilai_keypad
    End If
    If Power_motor > 100 Then Power_motor = 100
    Locate 2 , 11
    Lcd "    "
    Locate 2 , 11
    Lcd Power_motor ; " %"
End If
If Nilai_keypad = 10 Then
    Ulang = 0
    Writeeprom Temperature , Data_controller
    Writeeprom Mv_var
    Writeeprom Power_motor
End If
End If
Loop Until Ulang = 0
Goto Main_prog

```

Time_prog:

```

Readeeprom Temperature , Data_controller

```

```
Readeeprom Mv_var
Readeeprom Power_motor
Readeeprom Periode
Cls
Locate 1 , 1
Lcd "PROCESS DURATION"
Locate 2 , 1
Lcd "PERIODE=" ; Periode ; " Mn"
Ulang = 1
Iterasi = 0
Do
  Keypad = Getkbd()
  If Keypad < 16 Then
    Set Buzzer
    Select Case Keypad
      Case 2
        Nilai_keypad = 0
      Case 15
        Nilai_keypad = 1
      Case 14
        Nilai_keypad = 2
      Case 13
        Nilai_keypad = 3
      Case 11
        Nilai_keypad = 4
      Case 10
        Nilai_keypad = 5
      Case 9
        Nilai_keypad = 6
      Case 7
```

```

        Nilai_keypad = 7
    Case 6
        Nilai_keypad = 8
    Case 5
        Nilai_keypad = 9
    Case 1
        Nilai_keypad = 10
End Select
Waitms 100
Reset Buzzer
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
        Periode = Nilai_keypad
    Else
        Periode = 10 * Periode
        Periode = Periode + Nilai_keypad
    End If
    If Periode > 60000 Then Periode = 60000
    Locate 2 , 9
    Lcd "    "
    Locate 2 , 9
    Lcd Periode ; " Mn"
End If
If Nilai_keypad = 10 Then

```

```
    Ulang = 0
    Writeeprom Temperature , Data_controller
    Writeeprom Mv_var
    Writeeprom Power_motor
    Writeeprom Periode
  End If
End If
Loop Until Ulang = 0
Goto Main_prog

Time_base:
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
    Reset Heater3
  Else
    Power_heater = Power_heater - 1
    Set Heater1
    Set Heater2
    Set Heater3
  End If
  If Periode_1s = 100 Then
    Periode_1s_flag = 1
    Periode_1s = 0
    Power_heater = Mv_var
```

```
End If
End If
Return
```

```
Serial_in:
Disable Interrupts
Data_seri = Inkey()
If Data_seri = "*" Then
    Data_seri = Waitkey()
    If Data_seri = "G" Then Kirim_flag = 1
End If
Enable Interrupts
Return
```

```
Start_prog:
If Auto_flag = 1 Then
    Readeeprom Temperature , Data_controller
    Readeeprom Mv_var
    Readeeprom Power_motor
    Readeeprom Periode
    Start Adc
    Start Timer0
    Start Timer1
    Pwm1a = Power_motor
    Pwm1b = Power_motor
    Waktu = 60 * Periode
    Error = 0
    Sigma_error_kp = 0
```

```

Temp_lcd = 0
Kp = 204
Ti = 102
Td = 25.5
Cls
Locate 1 , 1
Lcd " AUTOMATIC MODE "
Locate 2 , 1
Lcd "SP=" ; Temperature ; Chr(223) ; "C"
Locate 2 , 10
Lcd "T=" ; Temp_lcd ; Chr(223) ; "C"
Do
  If Periode_1s_flag = 1 Then
    Periode_1s_flag = 0
    If Waktu = 0 Then
      Stop_flag = 1
    Else
      Waktu = Waktu - 1
    End If
    Error_old = Error
    Nilai_adc = Getadc(1)
    Nilai_temp = Nilai_adc * 0.466
    Nilai_temp = Round(nilai_temp)
    Temp_lcd = Nilai_temp
    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
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_var = Mv
Locate 2 , 10
Lcd "      "
Locate 2 , 10
Lcd Temp_lcd ; Chr(223) ; "C"
Keypad = Getkbd()
If Keypad = 1 Then Stop_flag = 1
End If
If Kirim_flag = 1 Then
  Kirim_flag = 0
  Print Temperature ; ":" ; Temp_lcd ; "#"
End If
Loop Until Stop_flag = 1
Stop Adc
Stop Timer0
Stop Timer1
Pwm1a = 0
Pwm1b = 0
Reset Heater1
Reset Heater2

```

Reset Heater3

Else

Readeeprom Temperature , Data_controller

Readeeprom Mv_var

Readeeprom Power_motor

Readeeprom Periode

Start Adc

Start Timer0

Start Timer1

Pwm1a = Power_motor

Pwm1b = Power_motor

Waktu = 60 * Waktu

Cls

Locate 1 , 1

Lcd " MANUAL MODE "

Locate 2 , 1

Lcd "MV=" ; Mv_var ; "%"

Locate 2 , 10

Lcd "T=" ; Temp_lcd ; Chr(223) ; "C"

Do

 If Periode_1s_flag = 1 Then

 Periode_1s_flag = 0

 If Waktu = 0 Then

 Stop_flag = 1

 Else

 Waktu = Waktu - 1

 End If

 Nilai_adc = Getadc(1)

 Nilai_temp = Nilai_adc * 0.466

 Nilai_temp = Round(nilai_temp)


```
Locate 2 , 12
Lcd "  "
Locate 2 , 12
Lcd Temp_lcd ; Chr(223) ; "C"
Keypad = Getkbd()
If Keypad = 1 Then Stop_flag = 1
End If
If Kirim_flag = 1 Then
  Kirim_flag = 0
  Print Mv_var ; ":" ; Temp_lcd ; "#"
End If
Keypad = Getkbd()
If Keypad = 1 Then Stop_flag = 1
Loop Until Stop_flag = 1
Stop Adc
Stop Timer0
Stop Timer1
Pwm1a = 0
Pwm1b = 0
Reset Heater1
Reset Heater2
Reset Heater3
End If
Goto Main_prog
```