



UNIVERSITAS INDONESIA

**RANCANG BANGUN ALAT PEMANAS DAN PENGADUK
TERINTEGRASI DENGAN TEMPERATUR DAN
KECEPATAN TERKENDALI BERBASIS
MIKROKONTROLER**

SKRIPSI

ADE MUNDARI WIJAYA

0906601935

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



UNIVERSITAS INDONESIA

**RANCANG BANGUN ALAT PEMANAS DAN PENGADUK
TERINTEGRASI DENGAN TEMPERATUR DAN
KECEPATAN TERKENDALI BERBASIS
MIKROKONTROLER**

SKRIPSI

Diajukan sebagai salah satu syarat meraih gelar sarjana sains

ADE MUNDARI WIJAYA

0906601935

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

HALAMAN PERNYATAAN ORISINALITAS

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

Nama : Ade Mundari Wijaya

NPM : 0906601935

Tanda Tangan : 

Tanggal : 15 Desember 2011

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :

Nama : Ade Mundari Wijaya
NPM : 0906601935
Program Studi : Ekstensi Fisika Instrumentasi
Judul Skripsi : Rancang Bangun Alat Pemanas dan Pengaduk Terintegrasi dengan Temperatur dan Kecepatan Terkendali Berbasis Mikrokontroler

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

DEWAN PENGUJI

Pembimbing 1 : Dr. Bambang Soegijono

Pembimbing 2 : Drs. Arief Sudarmaji, MT

Penguji 1 : Dr. Prawito

Penguji 2 : Lingga Hermanto, M.Si

Ditetapkan di : Ruang Seminar Fakultas MIPA, Kampus UI Depok

Tanggal : 15 Desember 2011

KATA PENGANTAR

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

Penyusunan Laporan Tugas Akhir yang berjudul “Rancang Bangun Alat Pemanas dan Pengaduk dengan Temperatur dan Kecepatan Terkendali” yang bertujuan untuk memenuhi syarat dalam menyelesaikan pendidikan S1 Departemen Fisika, program studi Instrumentasi, Fakultas MIPA, Universitas Indonesia. Dalam penyelesaian laporan ini banyak pihak yang telah berjasa, sehingga tanpa bantuan mereka pekerjaan ini tidak akan berarti. Tidak ada yang dapat saya persembahkan selain ucapan terima kasih yang terutama sekali kepada:

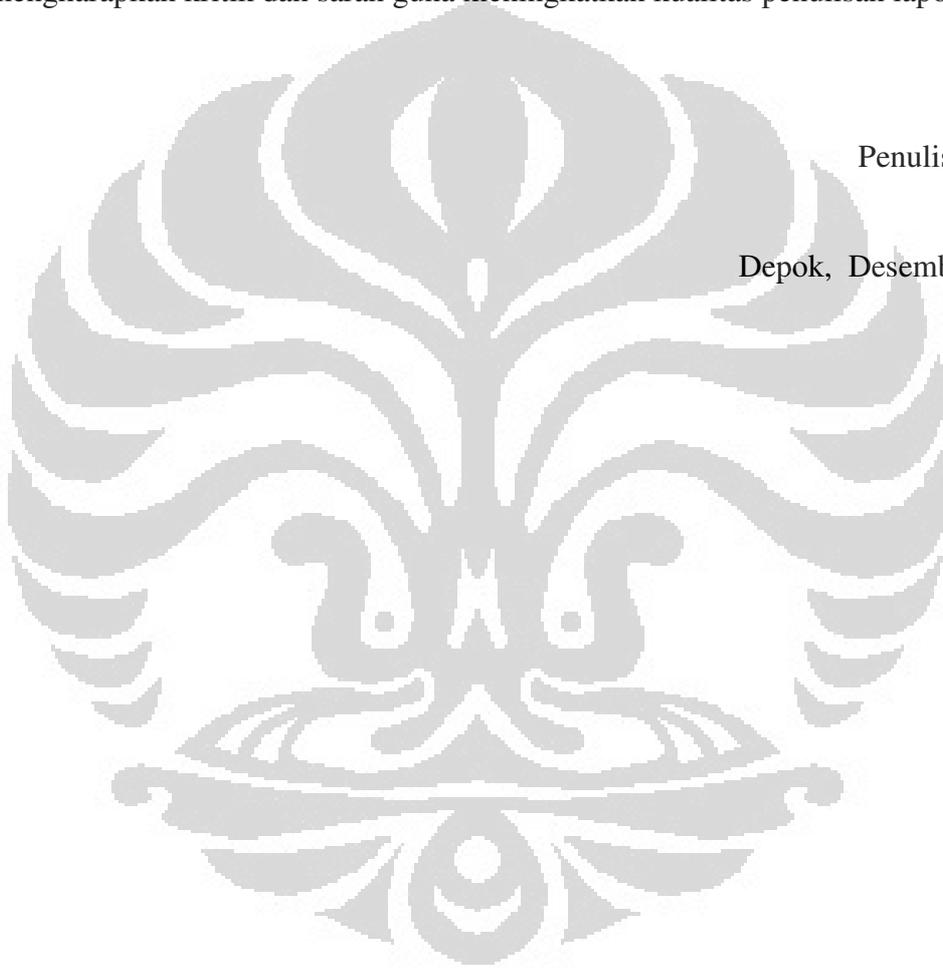
1. Kepada Istriku tercinta Freni Oktavia Hidayat yang telah memberikan semangat selalu menemani saat senang dan susah dalam penyelesaian tugas akhir ini.
2. Kepada kedua orangtua ku tercinta yang selalu mendukungku dari awal hingga selesainya tugas akhir ini.
3. Drs. Arief Sudarmaji, MT dan Dr. Bambang Soegijono selaku dosen pembimbing yang telah memberikan bimbingan dan pengetahuannya yang berkaitan dengan tugas akhir ini.
4. Teman – teman seangkatan yang selalu dapat bekerja sama dengan baik yaitu Fachrudin, Rahman, Misbah El Munir dan lain – lainnya yang tidak bisa aku sebutkan satu persatu.
5. Teman – temanku dikantor BPPT yaitu Roni Sujarwadi, Indah Fajarwati, Ibu Syuhada, dan teman – teman sekantor yang lain yang telah membantu

serta memberi semangat pantang menyerah untukku hingga selesainya laporan skripsi ini.

Kepada Allah SWT aku berdoa, mudah-mudahan, apa yang telah aku lakukan dan kemudian disusun laporan ini bermanfaat bagi para pembaca. Yang baik dan benar dan sempurna pastilah datangnya dari Allah SWT, dan segala yang tidak baik, yang tidak sempurna tentu merupakan ketidaksempurnaan ku sebagai hamba-Nya yang mempunyai banyak kekurangan. Untuk itu aku mengharapkan kritik dan saran guna meningkatkan kualitas penulisan laporan ini.

Penulis

Depok, Desember 2011



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

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

Nama : Ade Mundari Wijaya
NPM : 0906601935
Program Studi : Ekstensi Fisika Instrumentasi
Departemen : Fisika
Fakultas : Matematika dan Ilmu Pengetahuan Alam
Jenis karya : Skripsi

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 :

**RANCANG BANGUN ALAT PEMANAS DAN PENGADUK
TERINTEGRASI DENGAN TEMPERATUR DAN KECEPATAN
TERKENDALI BERBASIS MIKROKONTROLER**

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/format-kan, 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 : 15 Desember 2011

Yang menyatakan,

(Ade Mundari Wijaya)

ABSTRAK

Nama : Ade Mundari Wijaya
Program studi : Fisika Instrumentasi
Judul : Rancang Bangun Alat Pemanas dan Pengaduk Terintegrasi dengan Temperatur dan Kecepatan Terkendali

Telah dibuat alat pemanas dan pengaduk terintegrasi dengan temperatur dan kecepatan terkendali berbasis mikrokontroler dengan dilengkapi heater berdaya 600 W, 220 VAC dan sebagai sensor temperatur digunakan sensor temperatur termokopel tipe K yang memiliki *range* deteksi -200°C hingga 1200°C . Alat ini memiliki sebuah motor dc yang berfungsi sebagai pengaduk yang dilengkapi pula dengan sensor kecepatan putaran. Sebagai pengendali digunakan sebuah mikrokontroler ATmega8 yang digunakan sebagai pengontrol pemanas dan kecepatan putar motor pengaduk, dan disamping itu mikrokontroler juga menerima dan mengirimkan data ke komputer dengan software LabView atau keypad melalui komunikasi serial RS232 dimana setpoint dari LabView dan output dari mikrokontroler ditampilkan dalam bentuk nilai dan grafik. Pengaturan kecepatan putaran pengaduk dengan cara mengatur lebar pulsa (PWM), sedangkan untuk pengaturan panas heater menggunakan teknik kontrol PID dengan metode *Direct Synthesis*.

Kata kunci: *Heater*, Termokopel, Motor DC, PWM, PID

ABSTRACT

Name : Ade Mundari Wijaya
Study Program : Instrumentation of Physics, University of Indonesia
Title : Design of Integrated Heater and Stirrer with Temperature and Velocity Controlled by Using Microcontroller

Has created an integrated heater and stirrer with temperature and velocity controlled by using microcontroller, this instrument equipped by heater that has power about 600 W, 220 VAC. As temperature sensor used type K thermocouple where the detection range for this sensor $-200\text{ }^{\circ}\text{C}$ until $1200\text{ }^{\circ}\text{C}$. This instrument has a DC motor as stirrer and equipped by velocity sensor for velocity reading. As a controller was used a microcontroller ATmega 8 to control heater temperature and velocity of stirrer, beside that the microcontroller transmit and receive a number of data to and from computer through asynchronous serial communication RS232 with LabView software or keypad where setpoint from LabView and output from microcontroller displayed into a value and graph. The adjustment of velocity of stirrer rotation adjusted by changing pulsewidth of PWM, while the adjustment of heater temperature using PID control with direct synthesis method.

Key words: *Heater* , Thermocouple, DC Motor, PWM, PID

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PERNYATAAN ORISINALITAS	ii
HALAMAN PENGESAHAN.....	iii
KATA PENGANTAR	iv
LEMBAR PERSETUJUAN PUBLIKASI KARYA ILMIAH.....	vi
ABSTRAK	vii
DAFTAR ISI	ix
DAFTAR GAMBAR	xi
DAFTAR TABEL	xii
BAB 1 : PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Tujuan Penelitian	2
1.3 Deskripsi Singkat	2
1.4 Batasan Masalah	2
1.5 Metode Penelitian	2
1.6 Sistematika Penulisan	3
BAB 2 : TEORI DASAR	5
2.1 Sensor Temperatur Termokopel.....	5
2.2 Heater.....	8
2.3 Motor DC.....	9
2.3.1 Pengertian Motor DC.....	9
2.3.2 Mekanisme kerja Motor DC.....	11
2.3.3 Komponen Utama Motor DC.....	11
2.3.4 Jenis - Jenis Motor DC.....	13
2.4 Komunikasi Serial RS232.....	14
2.4.1 Konfigurasi Null Modem.....	17
2.4.2 Transmisi Data pada RS232.....	18
2.4.3 Keuntungan Menggunakan Komunikasi RS232.....	18
2.5 PWM (<i>Pulse Width Modulation</i>).....	19
2.6 PID (<i>Proportional Integral Differential</i>).....	21
BAB 3 : CARA KERJA SISTEM DAN PERANCANGAN ALAT..	27
3.1 Cara Kerja Sistem.....	27
3.1.1 Cara Kerja Subsistem Pengaduk.....	28
3.1.2 Cara Kerja Subsistem Pemanas.....	29
3.2 Perancangan Mekanik Alat.....	29
3.2.1 Perancangan Mekanik Heater.....	30
3.2.2 Perancangan Mekanik Box Rangkaian Utama.....	32
3.2.3 Perancangan Penjepit <i>Holder</i>	32
3.2.4 Perancangan Mekanik Box Keypad.....	32
3.3 Perancangan Hardware Rangkaian.....	33

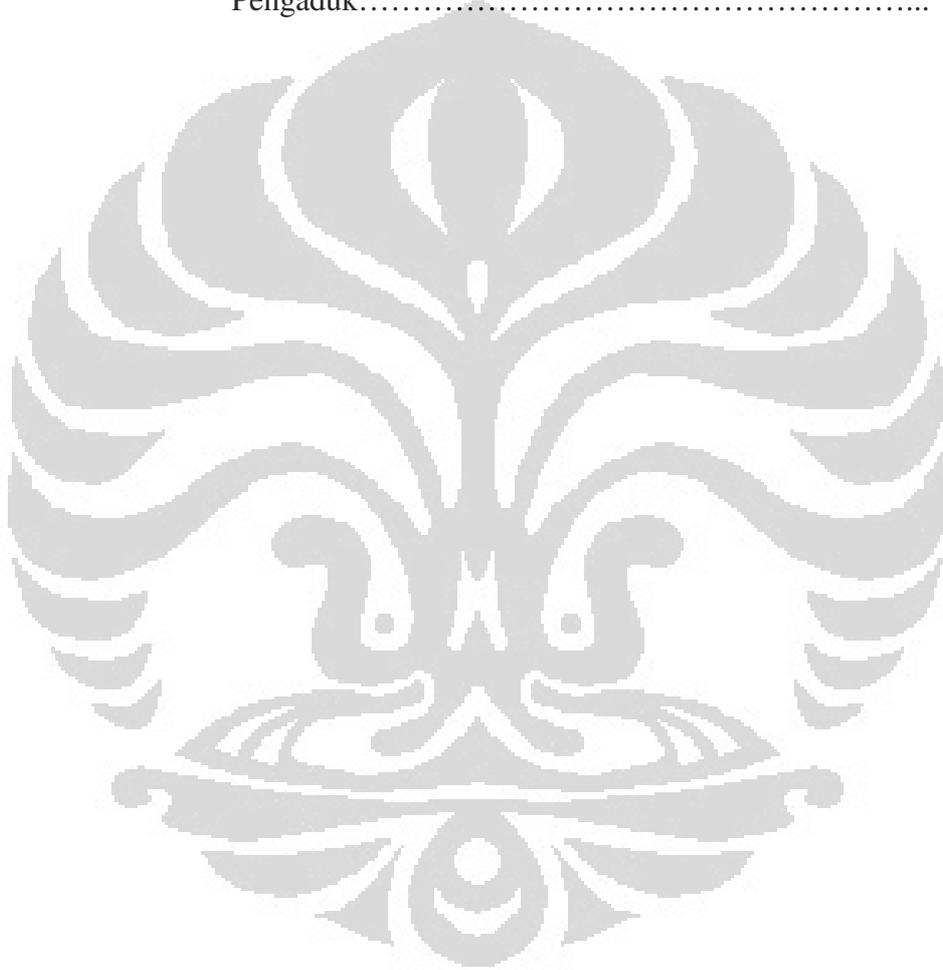
3.3.1 Perancangan <i>Driver</i> Motor DC.....	33
3.3.2 Perancangan Rangkaian <i>Cold Junction</i>	34
3.3.3 Perancangan Rangkaian Sistem Minimum ATMega 8 Untuk Rangkaian Utama.....	36
3.3.4 Perancangan Rangkaian Keypad.....	39
3.4 Perancangan Software Sistem.....	40
3.4.1 Perancangan Software untuk Mikrokontroler dan Keypad.....	40
3.4.2 Perancangan Software Labview.....	43
BAB 4 : PENGUJIAN DAN ANALISA DATA	44
4.1 Pengujian Rangkaian Keypad.....	44
4.2 Pengkalibrasian Rangkaian Pengkondisi Sinyal Termokopel (<i>Cold-Junction</i>)	45
4.3 Pengujian Rangkaian Driver Motor DC.....	47
4.4 Pengujian ADC Pada Mikrokontroler ATMega 8	49
4.5 Pengujian ADC termokopel vs Delta temperatur	51
4.6 Pengujian Sistem PID Dengan Metode <i>Direct</i> <i>Synthesis</i>	54
BAB 5 : KESIMPULAN DAN SARAN	57
5.1 Kesimpulan	57
5.2 Saran	57
DAFTAR PUSTAKA.....	58
LAMPIRAN	

DAFTAR GAMBAR

Gambar 2.1	Tegangan seebeck yang muncul akibat perubahan temperatur pada termokopel.....	6
Gambar 2.2	Bentuk Fisik Termokopel.....	6
Gambar 2.3	Kurva Respon Termokopel.....	7
Gambar 2.4	Bentuk Fisik termokopel yang digunakan	8
Gambar 2.5	Gaya lorentz sesuai kaidah tangan kanan.....	10
Gambar 2.6	Bentuk fisik bagian dalam motor DC.....	12
Gambar 2.7	Cara kerja motor DC.....	12
Gambar 2.8	Perbedaan Level Tegangan TTL dan RS232.....	15
Gambar 2.9	Konfigurasi Pin Pada Konektor DB9.....	16
Gambar 2.10	Konfigurasi Pengkabelan Pada Mode Null Modem...	18
Gambar 2.11	Sinyal PWM dengan Duty Cycle 50%.....	20
Gambar 2.12	Sistem Pengendali <i>Loop</i> Terbuka	22
Gambar 2.13	Sistem Pengendali <i>Loop</i> Tertutup	23
Gambar 2.14	Blok Diagram Pengendali <i>Proportional</i>	23
Gambar 2.15	Grafik respon Pengendali <i>Proportional</i>	24
Gambar 2.16	Blok Diagram Pengendali <i>Integral</i>	25
Gambar 2.17	Grafik Respon Pengendali <i>Integral</i>	25
Gambar 2.18	Blok Diagram Pengendali <i>Differensial</i>	26
Gambar 2.19	Grafik Respon Pengendali <i>Differensial</i>	26
Gambar 3.1	Diagram Blok Sistem Kontrol Alat.....	27
Gambar 3.2	Shaft Encoder Pada Motor DC.....	28
Gambar 3.3	Perencanaan Mekanik Sistem.....	30
Gambar 3.4	Heater <i>Hotplate</i>	31
Gambar 3.5	Perencanaan Mekanik Keypad.....	33
Gambar 3.6	Skematik Rangkaian <i>Driver</i> Motor DC.....	34
Gambar 3.7	Skematik Rangkaian <i>Cold Junction</i>	35
Gambar 3.8	Rangkaian Sistem Minimum ATmega 8.....	38
Gambar 3.9	Rangkaian PCB Utama yang Digabung Menjadi Satu	38
Gambar 3.10	Rangkaian Dasar Keypad.....	39
Gambar 3.11	Rangkaian PCB Keypad yang Telah Jadi.....	40
Gambar 3.12	Perancangan Software Bascom AVR untuk Heater dan Pengaduk.....	41
Gambar 3.13	Perancangan software Bascom AVR untuk keypad..	42
Gambar 3.14	Perancangan software untuk labview.....	43
Gambar 4.1	Grafik PWM Vs Kecepatan Rata – Rata.....	49
Gambar 4.2	Grafik hasil pengujian nilai ADC dengan range 0 mv hingga 51 mv.....	50
Gambar 4.3	Blok Diagram Labview.....	51
Gambar 4.4	Front Panel Labview.....	51
Gambar 4.5	Grafik Delta temperatur Vs nilai ADC termokopel pada saat temperatur turun dari 200°C hingga 35°C...	53
Gambar 4.6	Mencari Nilai θ dan Nilai τ	55
Gambar 4.7	Grafik hasil uji PID dengan setpoint 155 °C.....	56

DAFTAR TABEL

Tabel 2.1	Jenis – Jenis Heater	9
Tabel 2.2	Perbandingan fungsi pin DB9 dengan DB25.....	16
Tabel 2.3	Fungsi masing – masing pin.....	17
Tabel 4.1	Nilai keypad yang ditampilkan pada LCD 20x4 karakter	45
Tabel 4.2	Data Hasil Pengujian PWM Vs Kecepatan Putar Pengaduk.....	48



BAB 1

PENDAHULUAN

Pada saat ini teknologi semakin berkembang pesat sehingga banyak dibuat alat – alat yang menggunakan sistem otomatisasi tidak hanya di bidang industri melainkan dibidang – bidang yang lain, karena dengan sistem otomatisasi, kerja akan menjadi lebih efisien dan lebih produktif. Oleh karena itu untuk mencapai tujuan tersebut maka dikembangkanlah **“Alat Pengaduk dan Pemanas Terintegrasi dengan Temperatur dan Kecepatan Terkendali Berbasis Mikrokontroler”**.

1.1 Latar Belakang

Alat pemanas dan pengaduk adalah alat yang sangat sering digunakan pada kehidupan manusia sehari – hari. Biasanya pada kehidupan rumah tangga, pemanas diaplikasikan sebagai alat untuk memasak, menanak nasi, pemanas air dan sebagainya, sementara pengaduk banyak diaplikasikan sebagai alat pencuci pakaian, pengaduk adonan makanan dan sebagainya. Untuk cakupan yang lebih luas bahkan pemanas dan pengaduk menjadi hal yang sangat penting penunjang perekonomian, karena alat tersebut sangat banyak digunakan dalam industri, mulai dari industri makanan hingga industri pengepakan dan dari industri rumahan hingga industri besar seperti pabrik – pabrik makanan, minuman, maupun pengepakan. Namun disamping aplikasi alat tersebut terdapat teknik kontrol yang berbeda – beda pula yaitu ada yang terkendali secara manual, ada pula yang terkendali otomatis.

Saat ini telah banyak sistem pemanas dan pengaduk yang telah terkendali secara otomatis, tetapi pada aplikasinya masih jarang yang menggunakan software labview sebagai pengontrolnya. Oleh karena itu tujuan skripsi ini adalah untuk membuat atau merancang bangun sebuah sistem kendali untuk sistem pemanas dan pengaduk pada alat hotplate sehingga perubahan suhu dan kecepatan putaran pada pengaduk dapat dikontrol oleh sebuah komputer melalui software labview.

1.2 Tujuan Penelitian

Tujuan dilakukannya penelitian ini adalah membuat sistem pemanas dan pengaduk yang terkendali oleh mikrokontroler dan software labview sehingga didapatkan sebuah sistem terintegrasi.

1.3 Deskripsi Singkat

Alat yang akan dibuat adalah sebuah alat pemanas yang terdiri dari heater yang dilengkapi sensor pembaca suhu termokopel dan di atas alat ini terdapat sebuah motor dc yang berfungsi sebagai pengaduk yang dilengkapi juga dengan sensor kecepatan putaran. Sebagai pengendali digunakan sebuah mikrokontroler ATmega8 yang digunakan sebagai pengontrol pemanas dan kecepatan putar motor pengaduk , dan disamping itu mikrokontroler juga menerima dan mengirimkan data ke komputer atau keypad melalui komunikasi serial RS232 .

1.4 Batasan Masalah

Dalam penelitian ini, masalah yang akan diteliti dibatasi sesuai dengan judul yang diajukan yaitu “Rancang bangun alat pemanas dan pengaduk terkendali temperatur dan kecepatan berbasis mikrokontroler”. Sehingga penelitian ini hanya dibatasi pada: bagaimana merancang alat dan membuat program pada mikrokontroler dan software labview.

1.5 Metode Penelitian

Pada metode penelitian ini ada beberapa macam metode yang digunakan yaitu studi literatur, perancangan sistem, pembuatan sistem, pengujian sistem, pengambilan data dan analisa data.

1.5.1 Studi Literatur

Penulis menggunakan metode ini untuk memperoleh informasi dan data yang berkaitan dengan penelitian yang penulis lakukan. Studi literatur ini mengacu pada buku-buku pegangan, *data sheet* dari berbagai macam komponen yang dipergunakan, data yang didapat dari internet, dan makalah - makalah yang membahas tentang proyek yang terkait dengan judul yang akan penulis buat.

1.5.2 Perancangan Sistem

Membahas tentang desain dan cara kerja alat, perencanaan mekanik, sistem piranti elektronika dan lunak. Pada bagian piranti lunak akan dibahas program yang digunakan.

1.5.3 Pembuatan Sistem

Pembuatan sistem dilakukan sesuai dengan perancangan sistem yang telah di rancang. Pembuatan sistem dilakukan secara bertahap, mulai dari pembuatan mekanik, kemudian pembuatan piranti elektronik lalu perangkat lunak.

1.5.4 Pengujian Sistem

Dari sistem yang dibuat maka dilakukan pengujian secara menyeluruh, dengan tujuan untuk mengetahui apakah sudah sesuai dengan apa yang diharapkan atau belum.

1.5.5 Pengambilan Data

Pada bab ini akan diuraikan tentang kinerja dari percobaan dengan harapan dalam pengujian tidak terdapat kesalahan yang fatal.

1.5.6 Analisa Data

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 yang berkaitan dengan pengembangan atau modifikasi yang berkaitan dengan alat yang penulis buat.

1.6 Sistematika Penulisan

Sistematika penulisan skripsi 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 CARA KERJA SISTEM DAN PERANCANGAN ALAT

Pada bab ini akan dijelaskan secara keseluruhan tentang design dan cara kerja perencanaan mekanik, sistem piranti elektronika dan lunak. Pada bagian piranti lunak akan dibahas program yang digunakan.

BAB 4 HASIL EKSPERIMEN DAN ANALISA 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 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 bab ini akan dibahas semua teori yang mendukung perancangan alat yang mencakup sensor termokopel, *Heater*, motor DC, komunikasi serial RS232, PWM (*Pulse Width Modulation*), kontrol PID (Proporsional, Integral, Diferensial).

2.1. Sensor Temperatur Termokopel

D Sharon, dkk (1982), mengatakan sensor adalah suatu peralatan yang berfungsi untuk mendeteksi gejala-gejala atau sinyal-sinyal yang berasal dari perubahan suatu energi seperti energi listrik, energi fisika, energi kimia, energi biologi, energi mekanik dan sebagainya. Sensor secara umum didefinisikan sebagai alat yang dapat mengubah fenomena fisik kemudian mengubahnya menjadi sinyal elektrik baik arus listrik maupun tegangan. Fenomena fisik yang dapat menstimulasi sensor untuk menghasilkan sinyal elektrik meliputi temperatur, tekanan, gaya dan sebagainya. Sedangkan sensor itu sendiri terdiri dari transduser dengan atau tanpa penguat sinyal yang berbentuk dalam suatu sistem penginderaan jadi sebetulnya perbedaan antara sensor dan transduser adalah transduser adalah alat yang dapat mengubah energi dari satu bentuk ke bentuk yang lain dimana transduser tersebut mengubah energi listrik pada bentuk energi non listrik sedangkan sensor adalah alat yang dapat digunakan untuk mendeteksi. Sensor biasanya dikategorikan melalui pengukur dan memegang peranan penting dalam pengendalian proses pabrikasi modern.

Sensor memberikan ekivalen mata, pendengaran, hidung lidah dan menjadi otak mikroprosesor dari sistem otomatisasi industri. Sensor yang biasa ditemukan dan banyak diaplikasikan dalam industri adalah sensor temperatur. Temperatur adalah salah satu dari besaran fisika yang amat penting, temperatur didefinisikan sebagai ukuran relatif dari kondisi termal yang dimiliki suatu benda. Sensor temperatur adalah suatu transduser yang menangkap perubahan

temperatur menjadi suatu besaran fisika lain, seperti tegangan atau arus. Sensor temperatur yang digunakan dalam alat ini adalah termokopel. Berdasarkan prinsip seebeck yaitu menurut seorang fisikawan Estonia Thomas Johan Seebeck bahwa termokopel adalah sebuah detektor suhu yang didalamnya terdiri dari dua buah jenis logam yang berbeda dimana kedua ujung bahan logam yang berlainan tersebut akan menghasilkan beda potensial yang berubah - ubah yang dipengaruhi oleh perubahan temperatur lingkungannya (Gambar 2.1)



Gambar 2.1 tegangan seebeck yang muncul akibat perubahan temperatur pada termokopel[1]

Termokopel dapat mengukur temperatur dalam jangkauan yang besar, oleh karena itu termokopel menurut rentangnya dapat dibedakan menjadi beberapa tipe.

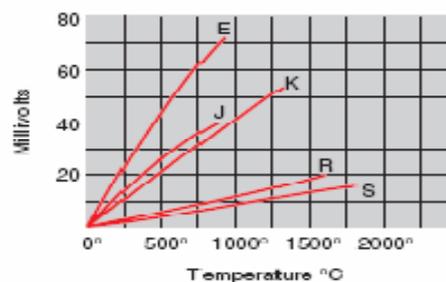


Gambar 2.2 Bentuk fisik termokopel[2]

National Institute of Standard and Technology (NIST) mempublikasikan berbagai macam jenis sensor termokopel berdasarkan jangkauan ukurnya menjadi beberapa tipe seperti dibawah ini:

1. Tipe K (*Chromel (Ni-Cr alloy) / alumel (Ni-Al alloy)*) termokopel ini digunakan untuk temperatur tinggi kira-kira $-200\text{ }^{\circ}\text{C}$ hingga $+1200\text{ }^{\circ}\text{C}$, dan harganya pun lebih murah dibanding tipe yang lain.

2. Tipe E (*Chromel / Constantan (Cu-Ni alloy)*), jenis ini memiliki output yang besar ($68\mu\text{V}/^\circ\text{C}$), digunakan untuk temperatur rendah.
3. Tipe J (*iron / Constantan*) rentangnya terbatas (-40 hingga $+7500^\circ\text{C}$), jenis ini memiliki sensitivitas sekitar $\sim 52\mu\text{V}/^\circ\text{C}$.
4. 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°C Sensitivitasnya sekitar $39\mu\text{V}/^\circ\text{C}$, sedikit di bawah tipe K.
5. Tipe B (*Platinum-Rhodium/Pt-Rh*) cocok untuk mengukur suhu diatas 1800°C .
6. 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.
7. Tipe S (*Platinum/Platinum with 10% Rhodium*) cocok untuk mengukur suhu dia atas 16000C , sensitivitasnya rendah ($10\mu\text{V}/^\circ\text{C}$), jenis termokopel ini pun harganya mahal.
8. Tipe T (*Copper/Constantan*) cocok untuk pengukuran antara -200 to 350°C , jenis ini memiliki sensitivitas $43\mu\text{V}/^\circ\text{C}$.



Gambar 2.3 Kurva Respon berbagai jenis termokopel [3]

Dalam pembuatan alat ini menggunakan termokopel tipe K, dimana termokopel tipe K ini memiliki karakteristik sebagai berikut:

- 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 pada 25°C adalah $40,6\mu\text{V}/^{\circ}\text{C}$



Gambar 2.4 Bentuk fisik termokopel yang digunakan

Perlu diingat bahwa termokopel tidak dapat mengukur temperatur absolut melainkan karena perbedaan temperatur diantara dua titik dan hubungan antara perbedaan suhu dengan tegangan yang dihasilkan termokopel bukan merupakan fungsi linier melainkan fungsi interpolasi polinomial[4]. Agar diperoleh hasil pengukuran yang akurat, persamaan biasanya diimplementasikan pada kontroler digital.

2.2 Heater

Heater adalah sebuah alat pemanas yang biasanya terbuat dari logam yang berupa lempengan, silinder pejal maupun berupa kawat pejal yang dibentuk menjadi spiral, sedangkan hotplate adalah sebuah pemanas yang berupa piringan yang didalam piringan tersebut terdapat elemen heater yang bisa berupa logam nichrome, tungsten atau lainnya, tetapi sering sekali digunakan sebagai pengganti salah satu pembakar dari berbagai oven atau bagian atas dari kompor masak. Hotplate atau piringan panas ini bisasanya sering digunakan untuk memanaskan makanan.

Secara umum terdapat berbagai macam jenis – jenis heater dapat ditemukan di dalam industri maupun dipasaran. Tabel 2.1 berikut ini memberikan informasi tentang jenis – jenis heater:

Tabel 2.1 Jenis – jenis heater

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

2.3 Motor DC

Motor DC memiliki pengertian, mekanisme kerja, komponen utama yang menyusun motor tersebut, serta beragam jenis - jenisnya.

2.3.1 Pengertian Motor DC

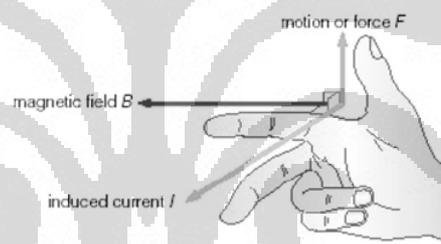
Motor DC merupakan sebuah perangkat elektromagnetis yang mengubah energi listrik menjadi energi mekanik. Energi mekanik ini digunakan untuk, misalnya, memutar impeller pompa, fan atau blower, menggerakkan kompresor, mengangkat bahan, dll. Motor listrik digunakan juga di rumah (mixer, bor listrik, fan angin) dan di industri. Motor listrik kadangkala disebut “kuda kerja” nya industri sebab diperkirakan bahwa motor-motor menggunakan sekitar 70% beban listrik total di industri.

Keuntungan utama motor DC adalah sebagai pengendali kecepatan, yang tidak mempengaruhi kualitas pasokan daya. Motor ini dapat dikendalikan dengan mengatur:

1. Tegangan dinamo – meningkatkan tegangan dinamo akan meningkatkan kecepatan
2. Arus medan – menurunkan arus medan akan meningkatkan kecepatan.

Motor DC tersedia dalam banyak ukuran, namun penggunaannya pada umumnya dibatasi untuk beberapa penggunaan berkecepatan rendah, penggunaan daya rendah hingga sedang seperti peralatan mesin dan rolling mills, sebab sering terjadi masalah dengan perubahan arah arus listrik mekanis pada ukuran yang lebih besar. Juga, motor tersebut dibatasi hanya untuk penggunaan di area yang bersih dan tidak berbahaya sebab resiko percikan api pada sikatnya.

Hubungan antara Gaya lorentz, fluks magnet, arus listrik, panjang penghantar dan sudut antaran arah arus dan fluks magnet adalah sebagai berikut:



Gambar 2.5 Gaya lorentz sesuai kaidah tangan kanan

$$F = B L I \sin \Theta$$

Dimana :

F = Gaya lorentz dalam satuan Newton (N)

B = kerapatan fluks magnet dalam satuan (weber / m²)

L = pajang penghantar dalam satuan meter (m)

I = arus listrik yang mengalir dalam satuan ampere (A)

Θ = sudut yang terbentuk antara arah medan magnet dengan arah arus yang mengalir pada kawat dalam satuan derajat (°)

2.3.2 Mekanisme Kerja Motor DC

Mekanisme kerja untuk seluruh jenis motor DC secara umum sama yaitu:

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

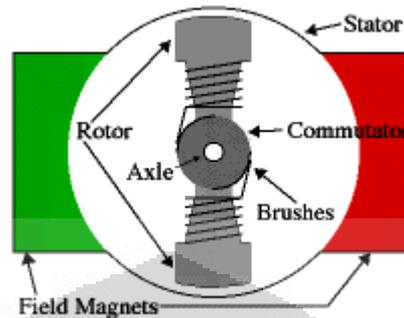
Dalam memahami sebuah motor, penting untuk mengerti apa yang dimaksud dengan beban motor. Beban mengacu kepada keluaran tenaga putar atau disebut juga dengan istilah *torque* sesuai dengan kecepatan yang diperlukan. Beban umumnya dapat dikategorikan kedalam tiga kelompok :

1. **Beban *torque* konstan** adalah beban dimana permintaan keluaran energinya bervariasi dengan kecepatan operasinya namun *torque* nya tidak bervariasi. Contoh beban dengan *torque* konstan adalah *conveyors*, *rotary kilns*, dan pompa *displacement* konstan.
2. **Beban dengan variabel *torque*** adalah beban dengan *torque* yang bervariasi dengan kecepatan operasi. Contoh beban dengan variabel *torque* adalah pompa sentrifugal dan fan (*torque* bervariasi sebagai kwadrat kecepatan).
3. **Beban dengan energi konstan** adalah beban dengan permintaan *torque* yang berubah dan berbanding terbalik dengan kecepatan. Contoh untuk beban dengan daya konstan adalah peralatan-peralatan mesin.

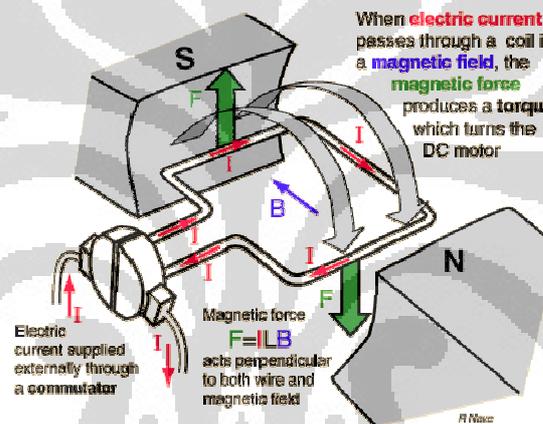
2.3.3 Komponen Utama Motor DC

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 Bentuk fisik bagian dalam motor DC[5]



Gambar 2.7 Cara kerja motor DC[6]

Sebuah motor DC yang memiliki tiga komponen utama (Gambar 2.6) :

1. **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.

2. **Dinamo.** Bila arus masuk menuju dinamo, maka arus ini akan menjadi elektromagnet. Motor yang berbentuk silinder, dihubungkan ke as penggerak untuk menggerakkan beban. Untuk kasus motor DC yang kecil, motor 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 motor.
3. **Commutator.** Komponen ini terutama ditemukan dalam motor DC. Kegunaannya adalah untuk membalikan arah arus listrik dalam motor. *Commutator* juga membantu dalam transmisi arus antara dinamo dan sumber daya.

2.3.4 Jenis – Jenis Motor DC

Pada dasarnya pembagian klasifikasi menurut sumber dayanya motor DC dibagi menjadi dua bagian yaitu:

a) Motor DC Sumber Daya Terpisah/ *Separately Excited*

Jika arus medan dipasok dari sumber terpisah maka disebut motor DC sumber daya terpisah/ *separately excited*.

b) Motor DC Sumber Daya Sendiri/ *Self Excited: motor shunt*

Pada motor *shunt*, gulungan medan (medan *shunt*) disambungkan secara paralel dengan gulungan motor. Oleh karena itu total arus dalam jalur merupakan penjumlahan arus medan dan arus motor.

Berikut tentang kecepatan motor *shunt* :

1. Kecepatan pada prakteknya konstan tidak tergantung pada beban (hingga *torque* tertentu setelah kecepatannya berkurang, lihat Gambar 4) dan oleh karena itu cocok untuk penggunaan komersial dengan beban awal yang rendah, seperti peralatan mesin.

2. Kecepatan dapat dikendalikan dengan cara memasang tahanan dalam susunan seri dengan dinamo (kecepatan berkurang) atau dengan memasang tahanan pada arus medan (kecepatan bertambah).

c) Motor DC daya sendiri: motor seri

Dalam motor seri, gulungan medan (medan *shunt*) dihubungkan secara seri dengan gulungan dinamo, oleh karena itu arus medan sama dengan arus dinamo. Berikut tentang kecepatan motor seri (Rodwell International Corporation, 1997; L.M. Photonics Ltd, 2002):

1. Kecepatan dibatasi pada 5000 RPM
2. Harus dihindarkan menjalankan motor seri tanpa ada beban sebab motor akan mempercepat tanpa terkendali. Motor-motor seri cocok untuk penggunaan yang memerlukan *torque* penyalaan awal yang tinggi, seperti derek dan alat pengangkat *hoist*.

d) Motor DC Kompon/Gabungan

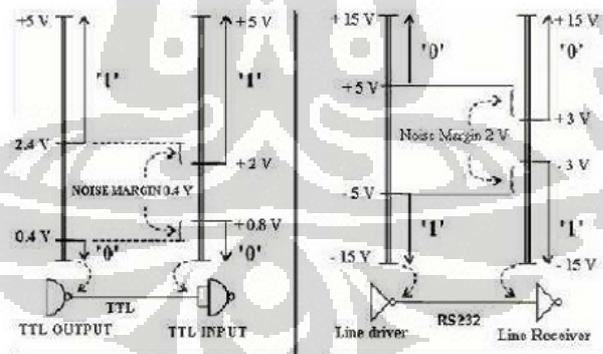
Motor Kompon DC merupakan gabungan motor seri dan *shunt*. Pada motor kompon, gulungan medan (medan *shunt*) dihubungkan secara paralel dan seri dengan gulungan. Sehingga, motor kompon memiliki *torque* penyalaan awal yang bagus dan kecepatan yang stabil. Makin tinggi persentase penggabungan (yakni persentase gulungan medan yang dihubungkan secara seri), makin tinggi pula *torque* penyalaan awal yang dapat ditangani oleh motor ini. Contoh, penggabungan 40-50% menjadikan motor ini cocok untuk alat pengangkat *hoist* dan derek, sedangkan motor kompon yang standar (12%) tidak cocok (myElectrical, 2005).

2.4 Komunikasi Serial RS232

Komunikasi data serial RS232 terdiri dari dua yaitu komunikasi data serial secara sinkron dan komunikasi data serial secara asinkron. Pada komunikasi data serial sinkron, clock dikirim bersama-sama dengan data serial, sedangkan komunikasi data serial secara asinkron, clock tidak dikirimkan bersama-sama dengan data serial, tetapi dibangkitkan secara sendiri-sendiri

baik pada sisi pengirim (transmitter) maupun pada sisi penerima (receiver). Pada IBM PC kompatibel port serialnya termaksud jenis asinkron. Komunikasi data serial ini dikerjakan oleh UART (Universal Asynchronous Receiver/ Transmitter). Pada UART, kecepatan pengiriman data (band rate) dan fase clock pada sisi transmitter dan pada sisi receiver harus sinkron. Sinkronisasi antar transmitter dan receiver dilakukan oleh bit 'Start' dan bit 'Stop'.

Standar RS232 ditetapkan oleh Electronic Industry Association and Telecommunication Industry Association pada tahun 1962. Nama lengkapnya adalah *EIA/TIA-232 Interface Between Data Terminal Equipment and Data Circuit-Terminating Equipment Employing Serial Binary Data Interchange*. Meskipun namanya cukup panjang tetapi standar ini hanya menyangkut komunikasi data antara komputer dengan alat-alat pelengkap komputer. Hal pokok yang diatur standar RS232, antara lain bentuk sinyal dan level tegangan yang dipakai. RS232 dibuat pada tahun 1962, jauh sebelum IC TTL populer, oleh karena itu level tegangan yang ditentukan untuk RS232 tidak ada hubungannya dengan level tegangan TTL, bahkan dapat dikatakan jauh berbeda. Berikut perbedaan antara level tegangan RS232 dan TTL :

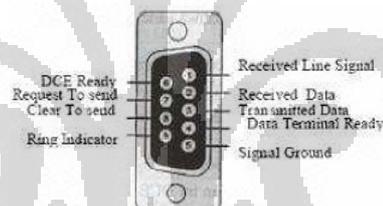


Gambar 2.8 Perbedaan tegangan TTL dan RS232

Penentuan jenis sinyal dan konektor yang dipakai, serta susunan sinyal pada kaki- kaki di konektor. Beberapa parameter yang ditetapkan EIA (Electronics Industry Association) antara lain:

1. Sebuah ‘spasi’ (logika 0) antara tegangan +3 s/d +25 volt
2. Sebuah ‘tanda’ (logika 1) antara tegangan -3 s/d -25 volt
3. Daerah tegangan antara +3 s/d -3 volt tidak didefinisikan
4. Tegangan rangkaian terbuka tidak boleh lebih dari 25 volt (dengan acuan ground)
5. Arus hubung singkat rangkaian tidak boleh lebih dari 500 mA.

Sebuah penggerak (driver) harus mampu menangani arus ini tanpa mengalami kerusakan. Selain mendeskripsikan level tegangan seperti yang dibahas di atas, standard RS232 menentukan pula jenis-jenis sinyal yang dipakai mengatur pertukaran informasi antara DTE dan DCE, semuanya terdapat 24 jenis sinyal tapi yang umum dipakai hanyalah 9 jenis sinyal. Konektor yang dipakai pun ditentukan dalam standard RS232, untuk sinyal yang lengkap dipakai konektor DB25, sedangkan konektor DB9 hanya bisa dipakai untuk 9 sinyal yang umum dipakai.



Gambar 2.9 Konfigurasi pin pada konektor DB9

Tabel 2.2 Perbedaan fungsi pin DB9 dan DB25

Pin DB25	Pin DB9	Singkatan	Keterangan
Pin 2	Pin 3	TD	<i>Transmit Data</i>
Pin 3	Pin 2	RD	<i>Receive Data</i>
Pin 4	Pin 7	RTS	<i>Request To Send</i>
Pin 5	Pin 8	CTS	<i>Clear To Send</i>
Pin 6	Pin 6	DSR	<i>Data Set Ready</i>
Pin 7	Pin 5	SG	<i>Signal Ground</i>
Pin 8	Pin 1	CD	<i>Carrier Detect</i>
Pin 20	Pin 4	DTR	<i>Data Terminal Ready</i>
Pin 22	Pin 9	RI	<i>Ring Indicator</i>

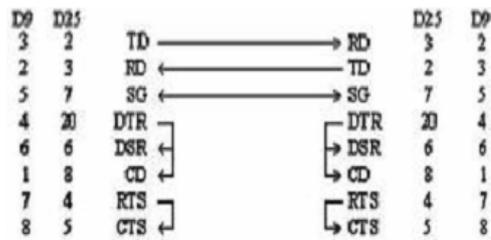
Tabel 2.3 Fungsi masing – masing pin RS232

Singkatan	Keterangan	Fungsi
TD	<i>Transmit Data</i>	Untuk pengiriman data serial (TDX)
RD	<i>Receive Data</i>	Untuk penerimaan data serial (RDX)
RTS	<i>Request To Send</i>	Sinyal untuk menginformasikan modem bahwa UART siap melakukan pertukaran data
CTS	<i>Clear To Send</i>	Digunakan untuk memberitahukan bahwa modem siap untuk melakukan pertukaran data
DSR	<i>Data Set Ready</i>	Memberitahukan UART bahwa modem siap untuk melakukan pertukaran data
CD	<i>Carrier Detect</i>	Saat modem mendeteksi suatu 'carrier' dari modem lain maka sinyal ini akan diaktifkan
DTR	<i>Data Terminal Ready</i>	Kebalikan dari DSR untuk memberitahukan bahwa UAT siap melakukan hubungan komunikasi
RI	<i>Ring Indicator</i>	Akan aktif jika modem mendeteksi adanya sinyal dering dari saluran telepon

Sinyal-sinyal tersebut ada yang menuju ke DCE ada pula yang berasal dari DCE. Bagi sinyal yang menuju ke DCE artinya DTE berfungsi sebagai output dan DCE berfungsi sebagai input, misalnya sinyal TD, pada sisi DTE kaki TD adalah output, dan kaki ini dihubungkan ke kaki TD pada DCE yang berfungsi sebagai input. Kebalikan sinyal TD adalah RD, sinyal ini berasal dari DCE dan dihubungkan ke kaki RD pada DTE yang berfungsi sebagai output.

2.4.1 Konfigurasi Null Modem

Konfigurasi Null Modem digunakan untuk menghubungkan dua DTE dengan diagram pengkabelan yang dapat dilihat pada gambar dibawah. Dalam hal ini hanya dibutuhkan tiga kabel antar DTE, yakni untuk TxD, RxD dan Gnd. Cara kerjanya adalah bagaimana membuat komputer agar berpikir bahwa komputer berkomunikasi dengan modem (DCE) bukan dengan komputer lainnya.



Gambar 2.10 Konfigurasi pengkabelan pada mode null modem

Pada gambar diatas terlihat bahwa kaki DTR (Data Terminal Ready) dihubungkan ke DSR (Data Set Ready) dan juga ke CD (Carrier Detect) pada masing masing komputer, sehingga pada saat sinyal DTR diaktifkan maka sinyal DSR dan CD juga ikut aktif (konsep Modem Semu atau Virtual Modem). Karena komputer dalam hal ini melakukan pengiriman data dengan kecepatan yang sama, maka kontrol aliran (flow control) belum dibutuhkan sehingga RTS (Request To Send) dan CTS (Clear to Send) pada masing masing komputer saling dihubungkan.

2.4.2 Transmisi Data Pada RS232

Komunikasi pada RS-232 dengan PC adalah komunikasi asinkron. Dimana sinyal clocknya tidak dikirim bersamaan dengan data. Masing-masing data disinkronkan menggunakan clock internal pada tiap-tiap sisinya. Format transmisinya satu byte pada RS232 dan data yang ditransmisikan pada format diatas adalah 8 bit, sebelum data tersebut ditransmisikan maka akan diawali oleh start bit dengan logika 0 (0 Volt), kemudian 8 bit data dan diakhiri oleh satu stop bit dengan logika 1 (5 Volt).

2.4.3 Keuntungan Menggunakan Komunikasi RS232

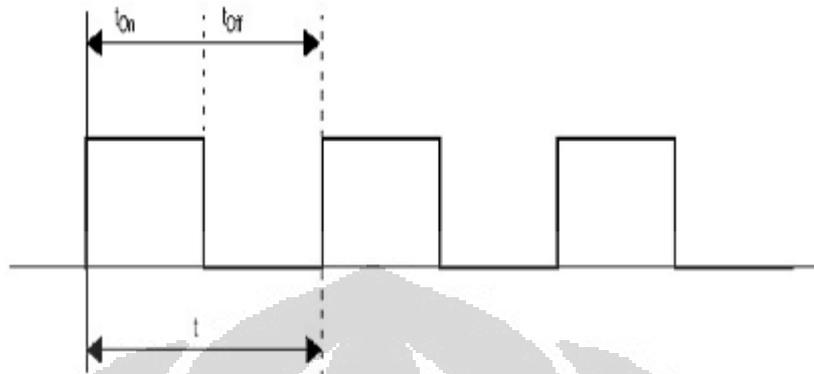
Antar muka komunikasi serial menawarkan beberapa kelebihan dibandingkan dengan komunikasi paralel, diantaranya:

1. Kabel untuk komunikasi serial bisa lebih panjang dibandingkan dengan paralel. Data-data dalam komunikasi serial dikirimkan untuk logika '1' sebagai tegangan -3 s/d -25 volt dan untuk logika '0' sebagai tegangan +3 s/d +25 volt, dengan demikian tegangan dalam komunikasi serial memiliki ayunan tegangan maksimum 50 volt, sedangkan pada komunikasi paralel hanya 5 volt. Hal ini menyebabkan gangguan pada kabel-kabel panjang lebih mudah diatasi dibanding dengan paralel.
2. Jumlah kabel serial lebih sedikit. Dua perangkat komputer yang berjauhan dengan hanya tiga kabel untuk konfigurasi null modem, yakni TxD (saluran kirim), RxD (saluran terima) dan Ground, akan tetapi jika menggunakan komunikasi paralel akan terdapat dua puluh hingga dua puluh lima kabel.
3. Komunikasi serial dapat menggunakan udara bebas sebagai media transmisi. Pada komunikasi serial hanya satu bit yang ditransmisikan pada satu waktu sehingga apabila transmisi menggunakan media udara bebas (*free space*) maka dibagian penerima tidak akan muncul kesulitan untuk menyusun kembali bit bit yang ditransmisikan.
4. Komunikasi serial dapat diterapkan untuk berkomunikasi dengan mikrokontroler. Hanya dibutuhkan dua pin utama TxD dan RxD (diluar acuan ground).

2.5 PWM (*Pulse Width Modulation*)

Suatu teknik yang digunakan untuk mengontrol kerja dari suatu alat atau menghasilkan suatu tegangan DC yang variabel adalah PWM. Rangkaian PWM adalah rangkaian yang lebar pulsa tegangan keluarannya dapat diatur atau dimodulasi oleh sinyal tegangan modulasi. Disamping itu kita dapat menghasilkan suatu sinyal PWM dengan menentukan frekuensi dan waktu dari variabel ON dan OFF. Pemodulasian sinyal yang beragam dapat

menghasilkan duty cycle yang diinginkan. Gambar 2.10 memperlihatkan sinyal kotak dengan duty cycle 50%.



Gambar 2.11 Sinyal PWM dengan duty cycle 50%

Duty cycle (lihat persamaan 2.3) adalah ratio dari waktu ON (t_{ON}) terhadap periode total dari sinyal ($t=t_{ON} + t_{OFF}$). Dengan persamaan :

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

Dimana :

D = Duty Cycle

t_1 = t_{ON}

t_2 = t_{OFF}

sehingga frekuensi yang dapat dihasilkan :

$$f = \frac{1}{T} \quad (2.4)$$

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

Modulasi lebar pulsa yang dihasilkan dari teknik PWM ini akan digunakan untuk mengatur kecepatan dari motor DC. Kemudian kita dapat mengetahui nilai RMS nya dengan menggunakan persamaan (2.3) dimana :

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

$$V_{RMS} = \sqrt{\frac{\int_0^{T_1} V(t)^2 dt}{T}}$$

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

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

$$V_{RMS} = \sqrt{\frac{[V_p^2 t]_0^t}{T}}$$

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

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

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

$$P = \frac{V_p^2}{R} \cdot \frac{t}{T} \quad (2.5)$$

2.6 PID (*Proportional Integral Differential*)

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*.

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

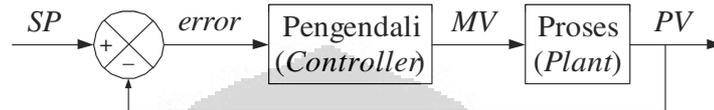
Open Loop Control System atau sistem pengendali *loop* terbuka merupakan sistem pengendalian dimana objek yang dikontrol tidak di-*feedback* ke pengendali, sehingga pengendali hanya akan memberikan output jika diberikan suatu sinyal input. Pengendali jenis ini masih bersifat manual karena tidak akan 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 2.12 menggambarkan sistem pengendali *loop* terbuka (*Open Loop Control System*).



Gambar 2.12 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 2.13 menggambarkan sistem pengendali *loop* tertutup (*Closed Loop Control System*).

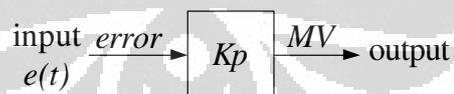


Gambar 2.13 Sistem Pengendali *Loop* Tertutup

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 2.14 Blok Diagram Pengendali *Proportional*

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

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

$$MV = K_p \cdot e(t) \quad (2.6)$$

$$K_p = \frac{1}{K} \cdot \frac{2\left(\frac{\tau}{\theta}\right) + 1}{2\left(\frac{\tau c}{\theta}\right) + 1} \quad (2.7)$$

$$K = \frac{\Delta PV}{\Delta MV} \quad (2.8)$$

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



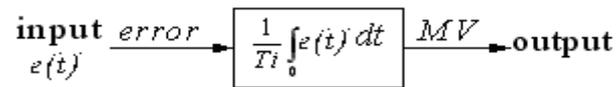
Gambar 2.15 Grafik respon Pengendali *Proportional*

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.

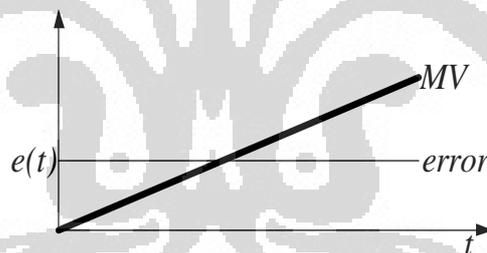
Gambar 2.16 Blok Diagram Pengendali *Integral*

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

$$MV = \frac{1}{Ti} \int_0^t e(t) dt \quad (2.9)$$

$$Ti = \frac{\theta}{2} + \tau \quad (2.10)$$

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.

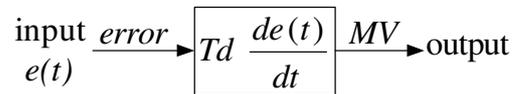
Gambar 2.17 Grafik Respon Pengendali *Integral*

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 *Differensial* (D)**

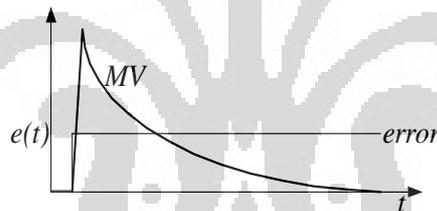
Pengendali *differensial* berfungsi untuk men-*differensial*-kan sinyal input lalu dikalikan dengan suatu besaran atau konstanta dengan nilai tertentu.



Gambar 2.18 Blok Diagram Pengendali *Differensial*

$$\mathbf{Td} = \frac{\tau}{2\left(\frac{\tau}{\theta}\right)+1} \quad (2.11)$$

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



Gambar 2.19 Grafik Respon Pengendali *Differensial*

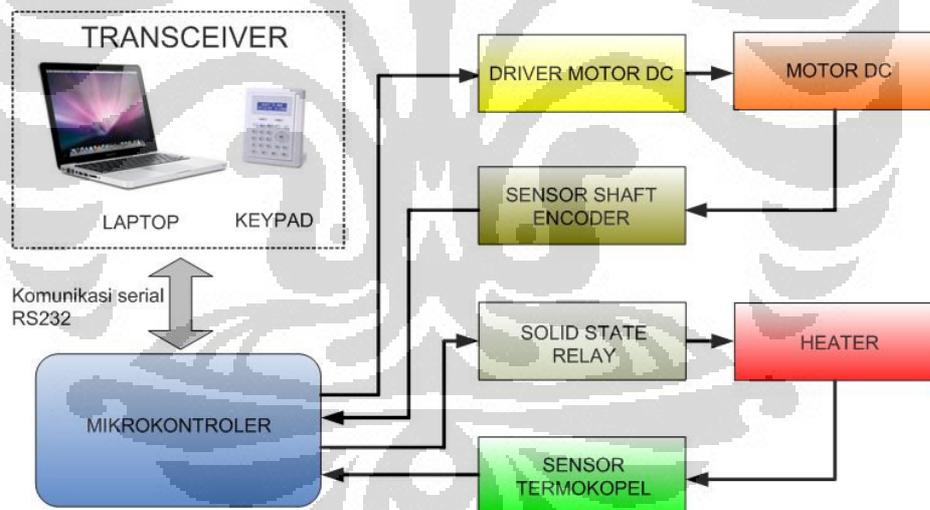
Pengendali *differensial* berfungsi untuk mengurangi respon yang terlalu berlebih yang dapat mengakibatkan *over shoot* pada proses *plant* karena nilai K_p yang terlalu besar pada pengendali *proportional*. Output dari pengendali *differensial* akan bernilai sangat besar jika perubahan *error* sangat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

BAB 3 CARA KERJA SISTEM DAN PERANCANGAN ALAT

Pada bab ini akan dibahas bagaimana sistem pada alat ini bekerja dan disamping itu akan dijelaskan perancangan mekanik dan elektrik pada alat ini mulai dari awal hingga akhir.

3.1 Cara Kerja Sistem

Alat ini dirancang untuk dapat mengendalikan temperatur pada *hotplate* dan kecepatan pada pengaduk sesuai keinginan penggunanya. Berikut ini (Gambar 3.1) adalah diagram blok cara kerja sistem pada alat:



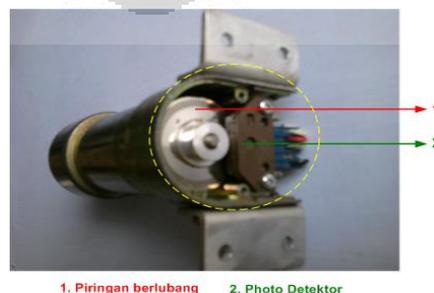
Gambar 3.1 Diagram Blok Sistem Kontrol Alat

Dari diagram blok di atas, keypad atau laptop berfungsi sebagai antarmuka yang menghubungkan operator dan alat untuk menginisialisasikan *microcontroller* yang bertindak sebagai pengendali, dimana sebagai *transceiver* yang mengirimkan nilai *Set Point* (SP) yang berupa bilangan dan

menampilkan nilai *Process Variable* (PV). Prinsipnya yang dikendalikan alat ini adalah kecepatan dan temperatur. Alat ini terdiri dari dua bagian utama atau subsistem yaitu subsistem pengaduk dan subsistem pemanas, sehingga ada dua bagian cara kerja yang berbeda diantara keduanya.

3.1.1 Cara Kerja Subsistem Pengaduk

Dalam pengendalian putaran pengaduk dengan motor DC ini, nilai dari Set Point yang berupa bilangan dari laptop atau keypad akan diolah oleh mikrokontroler, kemudian mikrokontroler akan mengeluarkan pulsa PWM (*Pulse Width Modulation*) yang berupa gelombang kotak. Pulsa PWM kemudian dikirim ke *driver* motor DC sehingga motor DC berputar dengan kecepatan putaran sesuai dengan nilai PWM yang diberikan oleh mikrokontroler. Besar nilai PWM berkisar 0 hingga 100% tergantung dari lebar pulsa yang telah ditentukan, sedangkan frekuensi yang digunakan ditetapkan 1 KHz karena motor DC yang dipakai untuk alat ini beroperasi dengan baik dengan nilai frekuensi tersebut. Kemudian putaran motor DC tersebut dibaca dengan menggunakan sensor putaran berupa piringan berlubang dimana lubang – lubang pada piringan tersebut dibaca oleh sensor cahaya (*Photo Detector*) (Gambar 3.2), kemudian sensor akan menghasilkan pulsa-pulsa untuk input counter dan nantinya akan kembali ke dalam sistem pengendali (mikrokontroler) berupa bilangan sebagai nilai Proses Variable (PV) lalu ditampilkan kembali oleh LCD pada keypad atau Laptop. Bagian pengendali kecepatan, yaitu bagian PWM dan counter terdapat di dalam mikrokontroler.



Gambar 3.2 Shaft Encoder pada Motor DC

3.1.2 Cara Kerja Subsistem Pemanas

Seperti halnya pengendalian kecepatan, pengendalian pemanas pada alat ini juga memiliki urutan kendali yang sama yaitu dimulai dari laptop atau keypad yang mengirimkan *set point* (SP) berupa bilangan ke mikrokontroler pada alat, kemudian mikrokontroler mengolah data tersebut setelah itu mikrokontroler mengendalikan SSR (*Solid State Relay*) agar *heater* beroperasi, disamping itu agar panas pada heater sesuai nilai yang diinginkan maka digunakan sensor panas termokopel tipe K yang dapat membaca suhu mulai dari $-200\text{ }^{\circ}\text{C}$ hingga $1200\text{ }^{\circ}\text{C}$. sensor membaca suhu yang dideteksi kemudian disinyal kondisikan oleh ADC (*Analog to Digital Converter*), data output ADC merupakan nilai *Manipulated Variable* (MV) yang menjadi nilai referensi untuk dibandingkan dengan *set point* kemudian mikrokontroler mengendalikan panas pada heater sehingga panasnya sesuai dengan yang diinginkan.

3.2 Perancangan Mekanik Alat

Alat ini terdiri dari beberapa bagian mekanik yaitu bagian mekanik heater, mekanik pengaduk, tiang *holder* dan mekanik untuk kontrol keypad (Gambar 3.3). Kotak heater terbuat dari *Stainless steel* dimana bahan tersebut tidak korosi jika terkena panas, begitu juga dengan tiang holder dan penjepit. Sedangkan untuk kotak rangkaian utama (bagian atas) dan *box* untuk keypad terbuat dari bahan *aluminium alloy* karena bahan tersebut cukup kuat untuk digunakan dan mudah dalam pembuatan lubang – lubang untuk *spacer* rangkaian. Pengaduk terbuat dari bahan karena terkait dengan cairan yang nanti pada aplikasinya adalah bahan – bahan kimia sehingga dengan bahan tersebut tidak bereaksi dengan cairan kimia tersebut. Untuk sensor yang digunakan adalah termokopel type K dimana sensor ini terbuat oleh *stainless steel* pula, bahan tersebut juga kuat terhadap bahan – bahan kimia.

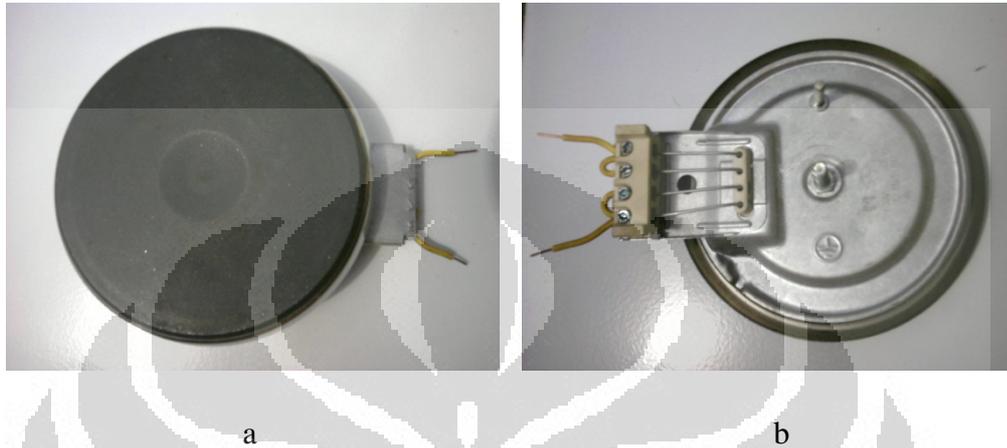


Gambar 3.3 Perencanaan mekanik sistem

3.2.1 Perancangan Mekanik Heater

Elemen pemanas (heater) yang digunakan untuk alat ini adalah sebuah heater *hotplate* dengan daya 600 watt (Gambar 3.4), dengan daya sebesar ini maka suhu maksimal yang dihasilkan apabila heater dibuat ON secara maksimal adalah 300°C, suhu ini sangat cukup untuk aplikasi dari alat pemanas ini di laboratorium. *Heater hotplate* ini sebenarnya banyak digunakan juga untuk kebutuhan pemanas rumah tangga, biasanya digunakan sebagai kompor listrik. *Hotplate* memiliki diameter lingkaran sebesar 150 mm, ukuran tersebut secara maksimal cukup untuk menampung *baker glass* ukuran 1000 ml atau 1 liter yang biasa digunakan untuk menampung atau wadah cairan – cairan bahan kimia. Untuk bagian mekanik atau box untuk heater, ukurannya disesuaikan dengan besarnya diameter heater dan ruang untuk kabel – kabel power, ukurannya adalah 250 x 200 x 40 mm (PxLxT).

Di dalam box bagian belakang heater terdapat sebuah bahan gipsum yang membatasi panas ketika heater beroperasi sehingga kabel – kabel yang ada dibelakang heater dapat terjaga.



Gambar 3.4 Heater hotplate

(a) Tampak atas (b) Tampak bawah

Pada aplikasi dalam kaitannya untuk kebutuhan rumah tangga, heater hotplate ini mempunyai beberapa keuntungan bila dibandingkan dengan pemanas kompor yang bersumber daya BBM atau BPG yaitu:

1. Lebih praktis, hal ini karena heater hotplate tidak memerlukan penggantian sumber daya seperti kompor biasa yaitu BBM, melainkan hanya butuh menghubungkannya dengan listrik 220 Volt dari PLN.
2. Lebih aman, karena pemanas ini tidak menggunakan tabung bertekanan sehingga bebas dari ancaman ledakan tabung seperti tabung elpiji.

Tetapi disamping itu terdapat kekurangannya juga, seperti ketika listrik dari sumber PLN padam maka alat tidak dapat beroperasi, atau dengan daya yang lumayan besar konsumsi listrik akan makin meningkat sehingga dapat

menyusahkan penggunaanya baik dalam hal biaya maupun dalam pengoperasiannya.

3.2.2 Perancangan Mekanik Box Rangkaian Utama

Ukuran mekanik pada box rangkaian disesuaikan dengan besarnya masing – masing alat yang ada didalamnya yang terdiri dari motor DC, rangkaian utama, solid state relay dan trafo, sehingga di pilihlah dimensi 165 x 105 x 55 mm (PxLxT). Sedangkan bahannya sendiri terbuat dari Aluminium alloy yang cukup kuat untuk menampung alat – alat yang ada didalamnya.

3.2.3 Perancangan Penjepit Holder

Pada bagian penjepit tiang, bahan yang digunakan adalah bahan stainless steel karena pada bagian ini penjepit harus cukup kuat atau tahan korosi sehingga box rangkaian yang di jepit tidak dapat jatuh setelah dikencangkan. Dimensi penjepit ini disesuaikan dengan tiang yang akan dijepit dengan diameter penjepit adalah 40 mm.

3.2.4 Perancangan Mekanik Box untuk Keypad

Bagian ini sama seperti box pada rangkaian utama baik dimensi maupun bahan yang terbuat. Namun box ini dirancang untuk menampung keypad 4x4 karakter dan konektor DB9 untuk komunikasi serial RS232 serta rangkaian keypad.



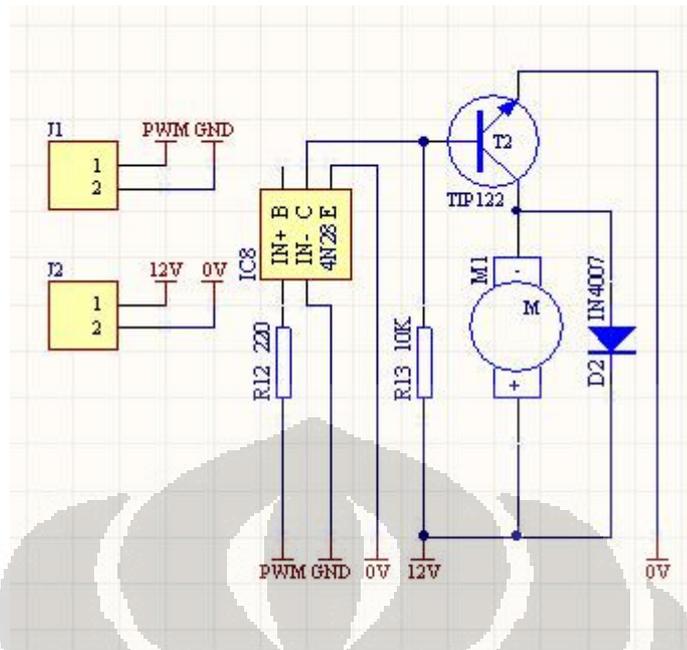
Gambar 3.5 Perencanaan Mekanik Keypad

3.3 Perancangan *Hardware* Rangkaian

Hardware rangkaian meliputi rangkaian *driver* motor DC, rangkaian pengendali temperatur yaitu rangkaian cold junction untuk termokopel, rangkaian sistem minimum dan rangkaian keypad.

3.3.1 Perancangan Driver Motor DC

Rangkaian ini harus mampu mengendalikan kecepatan motor DC dengan frekuensi yang diatur melalui PWM dan mampu melewati arus untuk motor DC yang besarnya 1.5A pada tegangan 12 V dan dalam keadaan tanpa beban oleh karena itu dipilih sebuah transistor dengan konfigurasi *darlington* sebagai penguat arus yaitu transistor NPN TIP122 dengan kemampuan melewati arus hingga 5A dan transistor dalam rangkaian ini digunakan sebagai saklar elektronik untuk mengatur PWM. untuk lebih jelas dapat dilihat gambar skema berikut (Gambar 3.6):



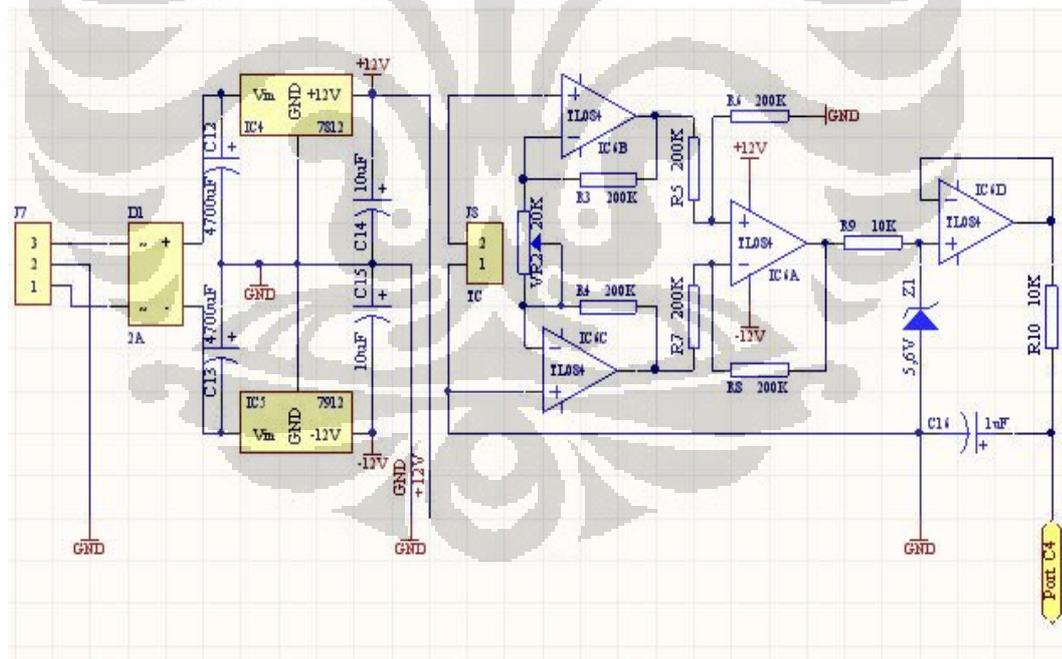
Gambar 3.6 Skematik Rangkaian Driver Motor DC

Dari rangkaian skematik diatas terlihat bahwa yang digunakan untuk *switching* dari tegangan digital (5VDC) ke tegangan yang lebih tinggi (12VDC) yaitu IC optocoupler tipe 4N28, karena dalam IC tersebut terdapat rangkaian photo isolator yaitu rangkaian *infra red* dan *photo detector* sehingga apabila terjadi short circuit pada tegangan 12V maka tidak menyebabkan kerusakan pada rangkaian yang menggunakan tegangan digital.

3.3.2 Perancangan Rangkaian Cold Junction

Rangkaian Cold Junction merupakan rangkaian untuk termokopel. Termokopel dihubungkan ke rangkaian cold junction melalui terminal input untuk termokopel. Rangkaian Op-Amp menggunakan TL084 dimana IC ini berbeda dengan IC Op-Amp pada umumnya yaitu tidak memiliki pin untuk mengatur tegangan offset (*Offsetnull*). Jadi karena tegangan offset tetap ada, maka saat mikrokontroler membaca tegangan dari sensor termokopel melalui ADC pada mikrokontroler ATmega 8, nilai ADC disaat suhu ruangan dalam kondisi suhu

ruangan pada umumnya yaitu sekitar 30°C , maka ditetapkan bahwa nilai ADC pada saat itu adalah nilai ADC tegangan offset sehingga pada saat pengukuran suhu dengan termokopel, semua output ADC harus dikurangi dengan nilai ADC tersebut sebagai pengganti kalibrasi. Kemudian untuk pembacaan suhu pada sensor LM35 adalah dimisalkan tegangan pada LM35 bernilai 250mV berarti suhu ruangan 25°C karena besar temperatur akan sama dengan besar tegangan dibagi dengan kenaikan $10\text{mV}/^{\circ}\text{C}$. Untuk mengkalibrasi termokopel, salah satu caranya adalah awalnya panaskan air hingga mendidih, lalu mencelupkan termometer dan termokopel ke air mendidih tersebut, kemudian misalnya suhu yang terbaca oleh termometer adalah 98°C sedangkan suhu lingkungan yang terbaca oleh LM35 adalah 30°C maka beda temperaturnya adalah $98^{\circ}\text{C} - 30^{\circ}\text{C} = 68^{\circ}\text{C}$ maka atur tegangan keluaran dari rangkaian *cold-junction* agar menjadi 680mv melalui hambatan Gain atau R_{gain} pada rangkaian *cold-junction*.



Gambar 3.7 Skematik Rangkaian Cold Junction

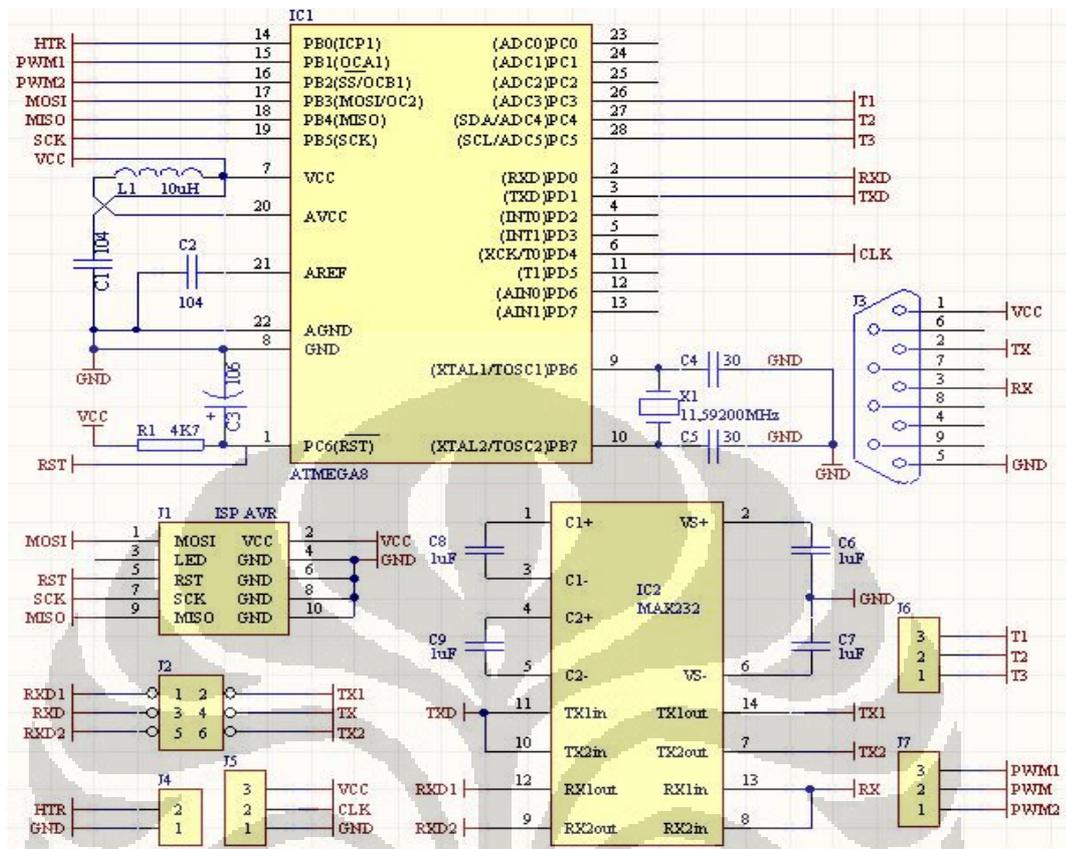
3.3.3 Perancangan Rangkaian Sistem Minimum ATmega 8 Untuk Rangkaian Utama

Mikrokontroler disini berfungsi sebagai protokol aktivitas pada Alat pemanas dan pengaduk serta sebagai pengontrol ADC. Sistem mikrokontroler yang direncanakan adalah menggunakan salah satu produk dari ATMEL yaitu ATmega8 yang mempunyai memori program internal 8 Kbyte dimana cukup untuk. Kelebihan dari jenis mikrokontroler ini adalah karena terdapat fasilitas ADC yang sudah built-in di dalamnya. Selain karena kelebihan tersebut, Mikrokontroler ATmega 8 dipilih juga karena diperkirakan memiliki harga yang ekonomis, tetapi tetap memiliki timer / counter dan memiliki ADC internal, serta jumlah port yang cukup walaupun hanya tersedia 3 port saja untuk digunakan dalam proyek akhir ini. Pada rangkaian di bawah, port - port yang dihubungkan dengan rangkaian downloader adalah port B.5 adalah SCK, port B.4 adalah MISO, port B.3 adalah MOSI, dan port C.6 adalah RESET.

Pada rangkaian sistem minimum mikrokontroler ATmega 8 dipakai frekuensi osilator (XTAL) sebesar 11.059.200 Hz. Hal ini akan menyebabkan agar memenuhi baudrate yang digunakan untuk komunikasi serial dengan menggunakan RS232 untuk memilih baudrate yang digunakan. Selain itu pada rangkaian ini disuplai oleh IC regulator 7805 sehingga tegangan catu yang dihasilkan stabil. Untuk mengatur jalannya data yang dikirim oleh komputer ke mikrokontroler agar data yang terkirim diterima dengan sempurna oleh mikrokontroler maupun data yang diterima oleh komputer dari mikrokontroler agar data yang terkirim diterima dengan sempurna oleh komputer diperlukan perangkat lunak. Sehingga pada rangkaian dibawah juga terdapat rangkaian pengubah tegangan IC MAX232. Rangkaian ini berfungsi untuk mengubah level keluaran tegangan yang keluar dari komputer yaitu level RS232 menjadi level tegangan TTL. Dimana tegangan pada level RS232 menjadi level logika "1" didefinisikan - 3V sampai -15V dan logika "0" didefinisikan +3V sampai +15V. Pada level TTL yang didefinisikan untuk kondisi "0" tegangannya 0 Volt sampai 0,4 Volt dan untuk kondisi "1" tegangannya 2,4 V sampai 5 V. Dalam perancangan ini untuk mengubah level tegangan

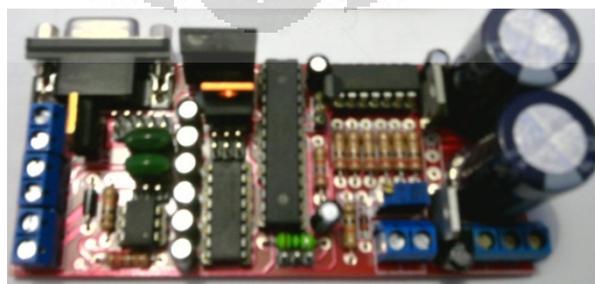
tersebut digunakan IC MAX232 dengan 4 buah kapasitor sebesar $1\mu\text{F}$ dengan tegangan catunya sebesar 5 volt. Dengan perangkat tersebut diharapkan dapat digunakan untuk mengirimkan atau menerima data dari komputer ke mikrokontroler atau dari mikrokontroler ke komputer dengan sempurna. Baudrate yang digunakan adalah 19200 Bps dengan mode asynchronous. Rangkaian mikrokontroler mempunyai kontrol direksi tiap bitnya yang dapat dikonfigurasi secara individual. Maka dalam pengkonfigurasi I/O yang digunakan ada yang berupa operasi port dan ada pula yang dikonfigurasi tiap bit I/O. Berikut ini akan diberikan konfigurasi dari I/O mikrokontroler yang berupa operasi port dan konfigurasi tiap bit yang ada pada masing-masing port yang terdapat pada mikrokontroler. Pada port C digunakan sebagai ADC untuk mengkonversi data yang dikirim oleh 2 buah data ke mikrokontroler yang digunakan sebagai sensor suhu pada lm 35 (port C3) untuk mengukur suhu lingkungan, lalu port C4 untuk mengukur output rangkaian cold junction.

Pada port B0 digunakan untuk mengontrol panas pada *hotplate* melalui rangkaian *Driver Solid state relay* sedangkan port B1 dan port B2 digunakan sebagai port kontrol PWM untuk penggerak kendali kecepatan motor DC melalui driver motor DC sedangkan port B6 dan port B7 digunakan untuk rangkaian *clock* atau Xtal. Port D hanya digunakan untuk komunikasi serial RS232 antara mikrokontroler ke komputer atau mikrokontroler ke keypad.



Gambar 3.8 Rangkaian Sistem Minimum ATmega 8

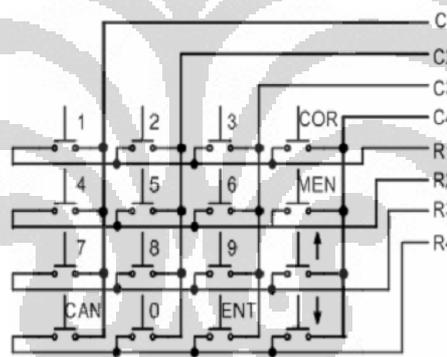
Pada akhirnya seluruh rangkaian yang termasuk dalam rangkaian utama dijadikan dalam satu buah desain PCB double layer, dipilih PCB double layer agar hasil rangkaian yang dihasilkan dapat memuat seluruh rangkaian dalam dimensi PCB yang kecil (Gambar 3.9)



Gambar 3.9 Rangkaian PCB utama yang digabung menjadi satu

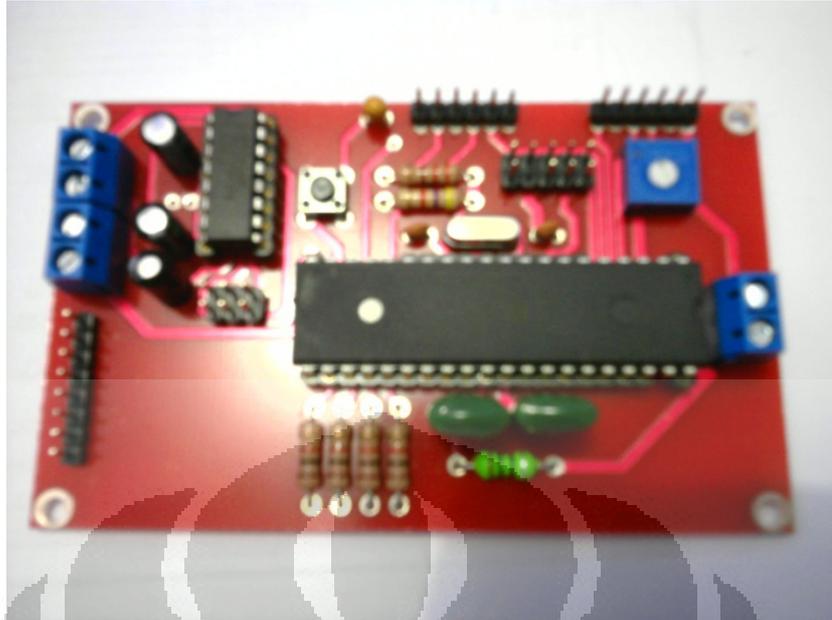
3.3.4 Perancangan Rangkaian Keypad

Keypad sering digunakan sebagai suatu input pada beberapa peralatan yang berbasis mikroprosesor atau mikrokontroler. Keypad sesungguhnya terdiri dari sejumlah saklar, yang terhubung sebagai baris dan kolom dengan susunan seperti yang ditunjukkan pada Gambar 3.10. Agar mikrokontroler dapat melakukan scan keypad, maka port mengeluarkan salah satu bit dari 4 bit yang terhubung pada kolom dengan logika low “0” dan selanjutnya membaca 4 bit pada baris untuk menguji jika ada tombol yang ditekan pada kolom tersebut. Sebagai konsekuensi, selama tidak ada tombol yang ditekan, maka mikrokontroler akan melihat sebagai logika high “1” pada setiap pin yang terhubung ke baris. Pada alat ini digunakan keypad 4 x 4 karakter.



Gambar 3.10 Rangkaian Dasar Keypad

Karena output pada rangkaian keypad harus mengirim data serial ke rangkaian utama, maka rangkaian keypad ini digabung sengan rangkaian sistem minimum. Rangkaian sistem minimum yang digunakan yaitu berbasis ATmega 16 dimana memori pada mikrokontroler ini lebih besar dibanding dengan rangkaian utama yang dilengkapi oleh ATmega 8, besar memorinya adalah 16 Kbyte, sehingga memudahkan untuk programmer membuat program menjadi lebih leluasa.



Gambar 3.11 Rangkaian PCB Keypad yang Telah Jadi

Dari gambar di atas tersedia juga rangkaian untuk LCD 20x4 yang ada di kotak rangkaian keypad.

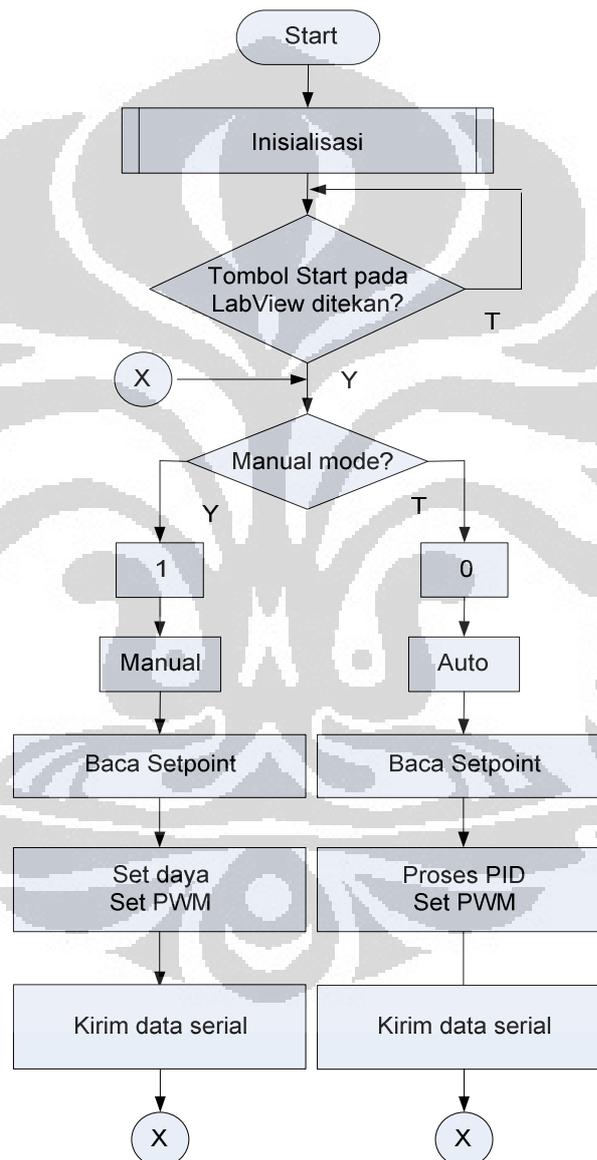
3.4 Perancangan Software Sistem

Software yang dirancang untuk alat ini terdiri dari software mikrokontroler dan software labview. Software pada mikrokontroler dibuat menggunakan Bascom AVR Versi 1.11.9.0 sedangkan software labview yang digunakan adalah versi 8.5.

3.4.1 Perancangan Software untuk Mikrokontroler dan Keypad

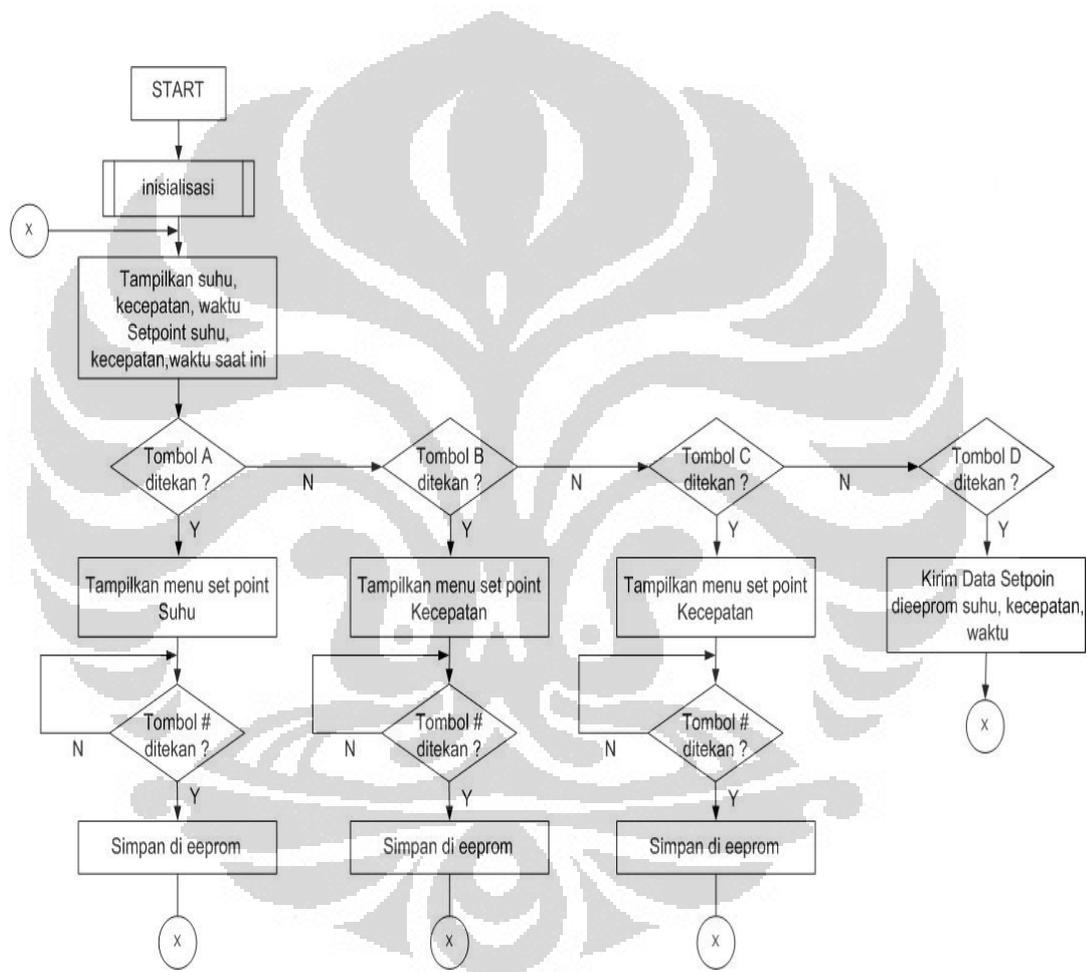
Software yang dirancang untuk mikrokontroler ATmega 8 harus mampu memonitor kecepatan motor dc, dapat menerima perintah dari software labview serta dapat mengirimkan data ke software labview melalui komunikasi serial RS232 dengan mode *Asynchronous*, sehingga software pada mikrokontroler harus

mampu melakukan interupsi serial, interupsi timer *overflow* oleh timer 2 yang digunakan sebagai pewaktu, pembangkit sinyal PWM oleh timer 1, penghitung kecepatan dengan menggunakan mode timer 0 sebagai *counter* serta mampu mengontrol heater dengan program PID. Perancangan software pada mikrokontroler ATmega 8 adalah sebagai berikut:



Gambar 3.12 Perancangan Software Bascom AVR untuk Heater dan Pengaduk

Semua rancangan software diatas (Gambar 3.12) adalah untuk pemrograman pada rangkaian heater dan pengaduk dengan menggunakan mikrokontroler ATmega 8, dimana menggunakan komunikasi interupsi serial dengan metode *Asynchronous*. . Karena software yang dirancang untuk heater dan pengaduk menggunakan komunikasi serial maka software yang dirancang untuk keypad maupun Labview juga harus menggunakan komunikasi serial. Berikut ini adalah rancangan software untuk keypad :

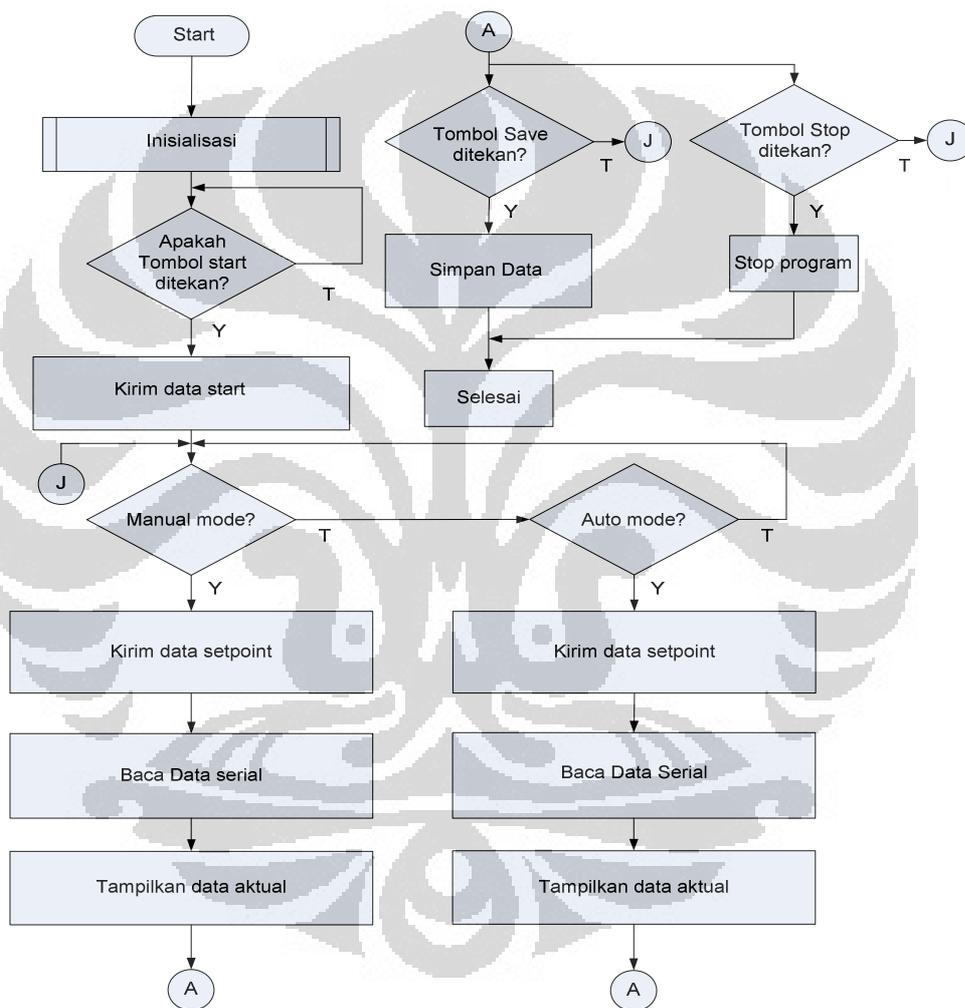


Gambar 3.13 Perancangan software Bascom AVR untuk keypad

Kemudian langkah selanjutnya adalah mengimplementasikan semua rancangan flowchart diatas menjadi program menggunakan software BASCOM AVR.

3.4.2 Perancangan Software Labview

Software yang dirancang untuk labview harus mampu melakukan komunikasi serial RS232 yang memiliki parameter – parameter yang sama dan mode pengiriman *baudrate* yang sama pula dengan setting yang ada di mikrokontroler baik dalam pengiriman maupun dalam penerimaan data.



Gambar 3.14 Perancangan software untuk labview

Kemudian langkah selanjutnya mengimplementasikan rancangan tersebut di atas menjadi Diagram blok pada software LabView.

BAB 4

HASIL EKSPERIMEN DAN ANALISA DATA

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

1. Pengujian Rangkaian keypad
2. Pengkalibrasian Rangkaian pengkondisi sinyal termokopel (*cold junction*)
3. Pengujian Rangkaian driver motor
4. Pengujian nilai temperatur termometer vs nilai ADC termokopel

4.1 Pengujian Rangkaian Keypad

Pengujian rangkaian keypad ini bertujuan ingin mengetahui rangkaian tersebut dapat berfungsi atau tidak. Rangkaian keypad ini terdiri dari mikrokontroler dan keypad matriks 4x4 dan LCD 20x4 karakter. Hasil pengujian ini dilakukan dengan memprogram rangkaian mikrokontroler dengan program keypad untuk membaca dan menampilkan nilai keypad ke LCD dengan bahasa *basic* BASCOM AVR Versi: 1.11.9.0. prosedur yang dilakukan adalah sebagai berikut:

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

Hasil yang didapatkan dari program tersebut yaitu seperti terlihat pada tabel 4.1 berikut ini:

Tabel 4.1 Nilai keypad yang ditampilkan pada LCD 20x4 karakter

No	Nomor / huruf pada tombol keypad	Nilai keypad pada LCD
1	1	0
2	2	1
3	3	2
4	A	3
5	4	4
6	5	5
7	6	6
8	B	7
9	7	8
10	8	9
11	9	10
12	C	11
13	*	12
14	0	13
15	#	14
16	D	15
17	Keypad tidak ditekan	16

Dengan hasil yang didapat seperti pada tabel di atas karena masing – masing tombol keypad mengeluarkan nilai maka dapat disimpulkan bahwa rangkaian keypad bekerja dengan baik.

4.2 Pengkalibrasian Rangkaian Pengkondisi Sinyal Termokopel (*Cold-Junction*)

Pengkalibrasian rangkaian cold junction ini bertujuan agar membuat respon antara termokopel dengan sensor suhu LM35 sama. Termokopel itu mengukur beda temperatur di ubah menjadi beda tegangan, maka perlu

diketahui beda temperatur dengan temperatur lingkung. Sebelumnya seharusnya rangkaian ini di offset nul agar tegangan kesalahan tidak ada. Tegangan offset keluaran (tegangan kesalahan) disebabkan oleh arus bias masukan. Idealnya bila tegangan kedua masukan sama besar, keluaran Op-Amp akan nol volt. Namun jarang ditemukan kejadian seperti itu sehingga keluarannya akan ada sedikit tegangan. Keadaan seperti inilah dapat di atasi dengan teknik membuat null tegangan offset. Namun karena tegangan yang digunakan DC dengan menggunakan op-amp cenderung lebih rentan terhadap perubahan temperatur yang dapat mengganggu keseimbangan op amp yang telah di atur sebelumnya akibatnya pada keluaran opamp akan terjadi kesalahan. Kesalahan disini berarti mempunyai beda tegangan.

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

1. Menghubungkan rangkaian termokopel ke sumber catu daya.
2. Mendownload program pembaca nilai ADC lm35 yang kemudian dikonversi menjadi temperatur. Nilai ini untuk mengukur temperatur lingkungan.
3. Menghubungkan sensor termokopel ke terminal termokopel.
4. Menghubungkan sensor lm35 ke terminal lm35.
5. Menghubungkan kabel *USB to Serial RS232 Converter* (disini digunakan merk ATEN) ke Notebook dan ke rangkaian
6. Memanaskan minyak goreng hingga 120 derajat pada wadah yang dicelupkan sensor termokopel dan termometer ke 1.
7. Membaca nilai ADC pada sensor LM35 kemudian dikonversi ke temperatur
8. Mengukur beda temperatur antara yang dihasilkan sensor lm35 dan temperatur yang terbaca pada termometer yang tercelup pada minyak goreng.

9. Mengukur tegangan keluaran rangkaian *cold-junction* hingga sama dengan beda temperatur dikalikan sepuluh dengan satuan mili volt.

Dari prosedur diatas dapat dideskripsikan dengan awalnya mengukur suhu lingkungan dengan sensor lm35, misal 30°C kemudian mengukur suhu pada minyak goreng misal 120 °C, kemudian selisih dari temperatur antara keduanya (90 °C) lalu dikalikan sepuluh (ketelitian sensor lm35 10mv/°C) sehingga hasilnya adalah 900mv, maka kemudian tegangan keluaran *cold-junction* harus bernilai 900mv dengan mengubah resistansi pada Rgain (trimpot) 20KΩ.

4.3 Pengujian Rangkaian *Driver Motor DC*

Pengujian ini bertujuan untuk mengetahui apakah driver motor DC dapat berfungsi dengan baik sebagai penggerak dan pembalik arah putaran dan untuk mengetahui kecepatan respon dari driver motor DC terhadap kecepatan perubahan arah putaran. Dalam pengujian ini hal yang perlu dilakukan adalah langkah pertama menyiapkan perlengkapan yang dibutuhkan dan kemudian langkah kedua merangkai seluruh perlengkapan menjadi sebuah rangkaian uji untuk driver motor:

- a). Langkah 1 : Menyiapkan Perlengkapan uji

Hal - hal yang perlu dipersiapkan dalam pengujian rangkaian driver yaitu:

1. Menyiapkan catu daya 12 Volt DC yang digunakan sebagai *power supply* untuk motor.
2. Menyiapkan catu daya 5 Volt DC yang digunakan sebagai *power supply* untuk *driver* motor yaitu IC 4n28, dimana IC ini dapat memisahkan antara ground motor dengan ground pengendali, hal ini membuat rangkaian menjadi lebih aman ketika beroperasi. IC 4n28 ini adalah sebuah *opto coupler* dimana input merupakan rangkaian *photoemitter* yang dapat mengeluarkan cahaya apabila diberi tegangan input yaitu 5 Volt DC, sedangkan output beroperasi akibat adanya cahaya yang terdeteksi oleh *phototransistor* yang basisnya akan aktif jika ada cahaya. Ketika basis

aktif maka mengakibatkan kolektor dan emiter akan terhubung, sehingga motor akan berputar.

3. Menyiapkan motor DC yang akan diuji.

b.) Langkah 2: Menghubungkan Seluruh Perlengkapan uji dan Mengoperasikan rangkaian driver motor DC.

Hal – hal yang perlu dilakukan untuk mengoperasikan rangkaian driver ini adalah:

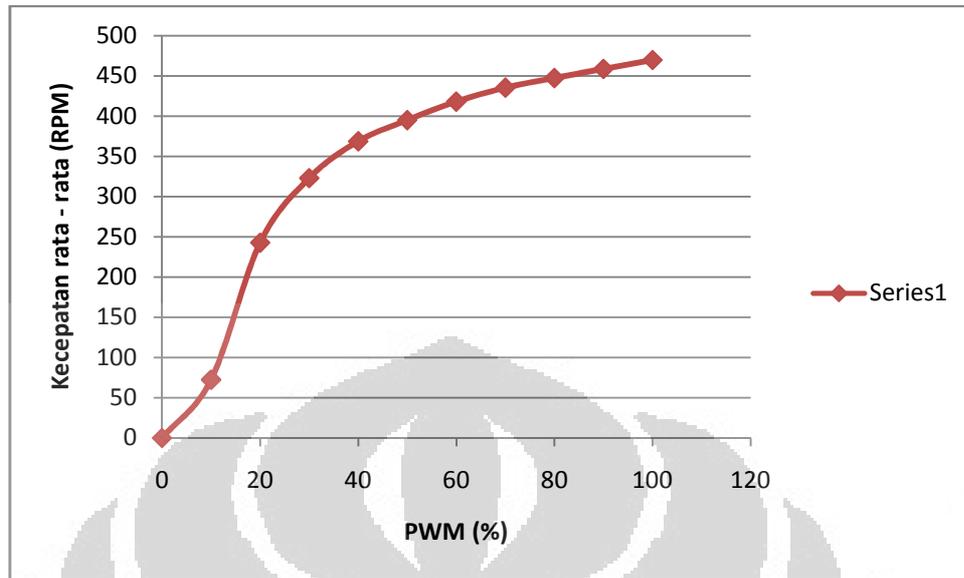
1. Menghubungkan kabel motor DC ke terminal output driver motor DC
2. Menghubungkan kabel termokopel ke terminal termokopel.
3. Menghubungkan kabel hotplate ke terminal hotplate.
4. Menghubungkan kabel serial RS232 ke terminal RS232 yang tersedia.
5. Menghubungkan sumber catu daya 12 Volt DC ke terminal input motor DC.
6. Menghubungkan sumber catu daya AC ke terminal input.
7. Rangkaian siap dioperasikan sesuai program yang diinginkan.

Dari kedua langkah diatas motor DC beroperasi dengan baik dengan hasil pengujian PWM sebagai berikut:

Tabel 4.2. Data Hasil Pengujian PWM Vs Kecepatan Putar Pengaduk

No	PWM (%)	Kecepatan (RPM)			Kecepatan Rata - Rata (RPM)
1	0	0	0	0	0
2	10	72	72	73	72.3
3	20	243	242	243	242.7
4	30	323	322	324	323.0
5	40	368	369	369	368.7
6	50	394	395	396	395.0
7	60	417	418	419	418.0
8	70	434	436	436	435.3
9	80	448	447	447	447.3
10	90	458	459	459	458.7
11	100	470	470	469	469.7

Grafik dari data diatas adalah sebagai berikut:



Gambar 4.1. Grafik PWM Vs Kecepatan Rata – Rata

Data pada grafik di atas didapat dari hasil pengukuran dengan menggunakan alat ukur **Digital Photo Tachometer Merk Dekko Tipe 2234L**, dari grafik diatas terlihat bahwa kecepatan putaran motor kurang linier, hal ini dimungkinkan karena kondisi fisik motor yang memiliki *gearbox* sehingga ada daerah mati atau *dead band* yang dimiliki gearbox tersebut dan hal lain yang memungkinkan adalah bahwa semua komponen dalam gearbox serta motor dc tersebut tidak dalam kondisi baru, tetapi hal ini tetap tidak bermasalah karena yang diperlukan adalah kekonstanan kecepatan motor disetiap level PWM.

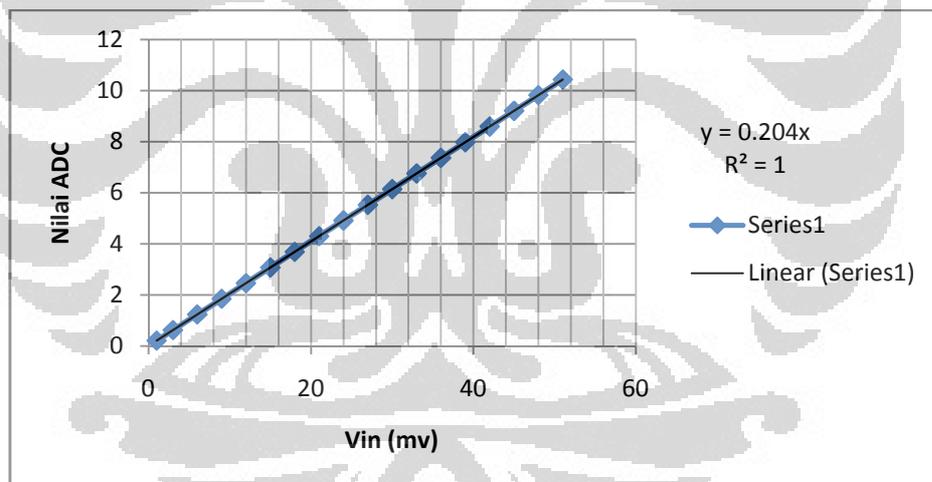
4.4 Pengujian ADC Pada Mikrokontroler ATmega 8

Pengujian ini bertujuan untuk mengetahui apakah nilai ADC yang ditampilkan mikrokontroler linier atau tidak. Untuk tujuan tersebut maka dilakukan prosedur pengujian sebagai berikut:

- a. Menghubungkan seluruh rangkaian mikrokontroler ATmega 8

- b. Menyiapkan tegangan input variabel dari 0 hingga 51 mv dc dengan menaikkan level tegangan setiap 3 mv (hal ini ditentukan oleh penulis agar data tidak terlalu banyak) yang kemudian dihubungkan ke salah satu terminal ADC.
- c. Mendownload program untuk membaca nilai ADC ke mikrokontroler.
- d. Menghubungkan kabel serial RS232 dari mikrokontroler ke notebook.
- e. Menampilkan nilai ADC yang terukur ke LabView dengan menghidupkan rangkaian pada mikrokontroler terlebih dahulu kemudian menekan tombol start pada program LabView.
- f. Mengubah – ubah tegangan input dari 0 mv hingga 51 mv.
- g. Menyimpan data hasil pengujian ke format excel (.xls)

Dari hasil pengujian didapat nilai ADC seperti grafik berikut ini:

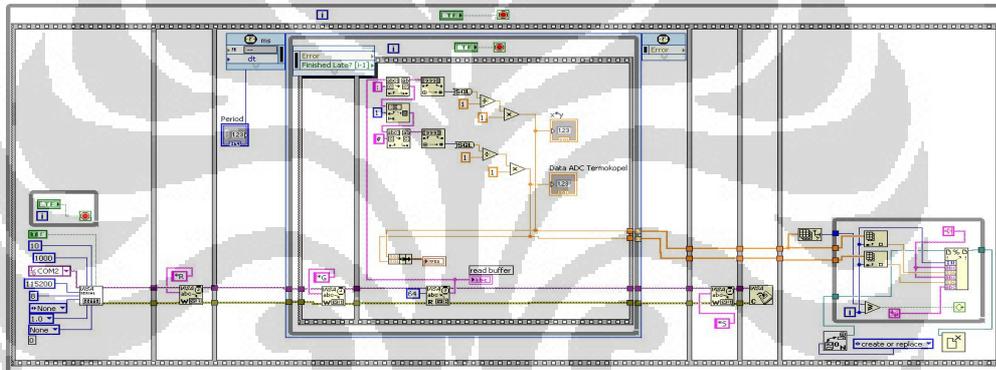


Gambar 4.2 Grafik hasil pengujian nilai ADC dengan range 0 hingga 51 mv

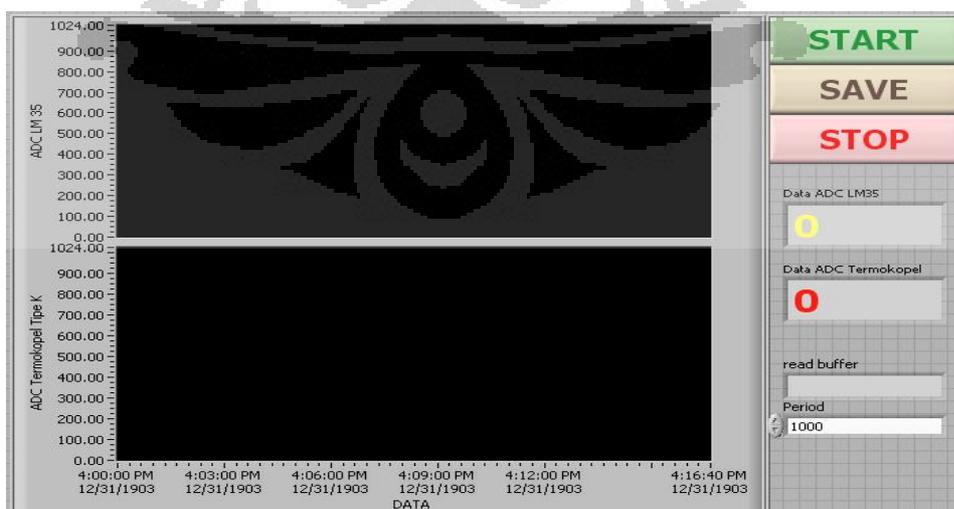
Dari grafik didapatkan informasi bahwa nilai ADC linier terhadap tegangan input dan indikator lain bisa dilihat pada nilai R^2 (koefisien korelasi) bernilai 1 yang artinya grafik yang dihasilkan linier.

4.5 Pengujian ADC termokopel vs Delta temperatur

Pengujian ini bertujuan untuk mendapatkan persamaan garis sehingga didapatkan faktor pengali untuk mengubah nilai ADC termokopel menjadi nilai temperatur untuk kemudian digunakan pada program penampil nilai temperatur pada termokopel. Pengujian ini menggunakan program labVIEW dengan cara, rangkaian pengendali yaitu avr dengan IC Atmega8 dihubungkan dengan komputer menggunakan serial RS232. Rangkaian ini dapat mengirimkan data serial ke komputer karena terdapat IC max 232. Gambar diagram blok yang digunakan pada pengambilan data dapat dilihat pada gambar 4.2



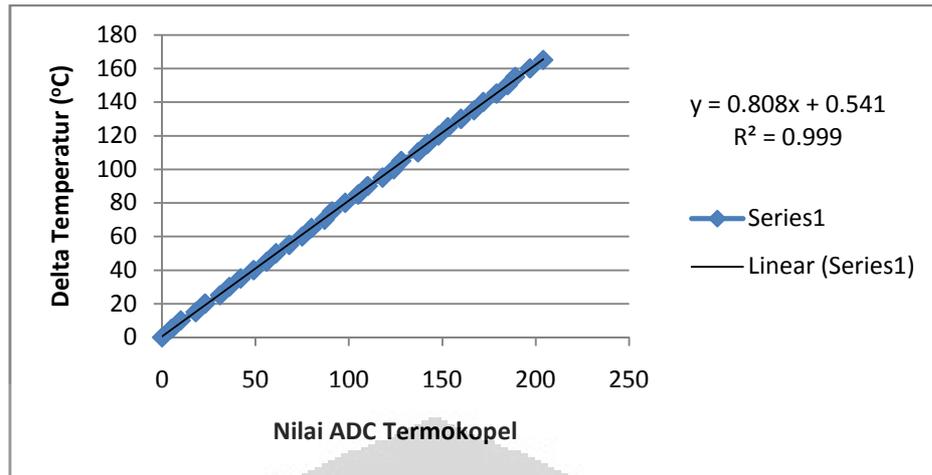
Gambar 4.3 Blok Diagram Labview



Gambar 4.4 Front Panel LabView

Secara singkat cara pengambilan data ADC adalah setelah rangkaian cold junction di kalibrasi. Pengukuran dilakukan dengan menggunakan termometer digital, dimana termokopel di celupkan ke minyak goreng yang di panaskan hingga 200°C, kemudian program pada labview di mulai dengan menekan tombol “START” kemudian data ADC termokopel disimpan secara otomatis oleh labview dengan file berformat .XLS (Excel), sedangkan temperatur di ukur dan dicatat manual setiap penurunan temperatur sebesar 5°C.

Penjelasan secara rinci adalah sebagai berikut, termokopel dihubungkan ke termometer digital dan rangkaian cold junction. Rangkaian cold junction di beri masukan dengan ± 12 volt. Tegangan keluaran di *cold-junction* dihubungkan ke rangkaian sistem minimum pada ADC. Rangkaian sistem minimum pada komunikasi serial di hubungkan ke PC melalui DB-9 dan USB serial. Program LabView dijalankan, kemudian mencatat temperatur sesuai dengan yang terbaca pada termometer digital misal 200°C setelah selesai mencatat semua data maka selanjutnya mencatat delta temperatur antara data suhu yang tercatat dengan suhu ruangan. Pada front panel labview klik start dengan mengklik *icon* berwarna merah untuk mengaktifkan agar program labview siap di eksekusi. Setelah program siap dieksekusi kemudian langkah selanjutnya mengklik tombol start yang berwarna hijau di *front panel* labview, sehingga program mulai merekam data yang terukur dengan dimulai dari temperatur 200°C hingga 35 °C. jika pengujian telah mencapai 35°C kemudian mengklik tombol save pada *front panel* labview, dan data langsung tersimpan dengan format Ms.Excel (.xls). Setelah melakukan percobaan didapatkan nilai ADC:



Gambar 4.5 Grafik Delta temperatur Vs nilai ADC termokopel pada saat temperatur turun dari 200°C hingga 35 °C

Nilai ADC pada grafik tersebut telah dikurangi nilai ADC pada temperatur ruang (71) karena Op-Amp yang digunakan tidak memiliki pengaturan offset null. Dari percobaan yang telah dilakukan di dapat bahwa pada saat penurunan temperatur, persamaan yang di dapat adalah linier. Hal ini terjadi karena pada saat mengkalibrasi termokopel dilakukan dengan menggunakan termometer digital yang diletakkan sedekat mungkin dengan termokopel sehingga membuat hasil kalibrasi semakin akurat, dan pengambilan data dengan menggunakan program labview, dimana pengambilan data ini memperoleh data sebanyak 34 buah data karena data yang diambil setiap penurunan 5°C. Dari grafik terlihat bahwa perubahan temperatur linier dengan persamaan garis yaitu:

$$Y = 0.808X + 0.541 \quad (4.1)$$

$$\text{Dengan } R^2 = 0.999$$

Dimana Y adalah nilai temperatur termokopel yang kemudian akan dihitung dari nilai dari ADC (sumbu X). Dari persamaan garis didapatkan nilai R^2

(Koefisien Korelasi) = 0,999 yang artinya sensor temperatur yang digunakan dalam pengukuran dapat dikatakan baik. Data yang diperoleh ini bisa memudahkan dalam mendapatkan nilai persamaan yang akan dimasukkan dalam program.

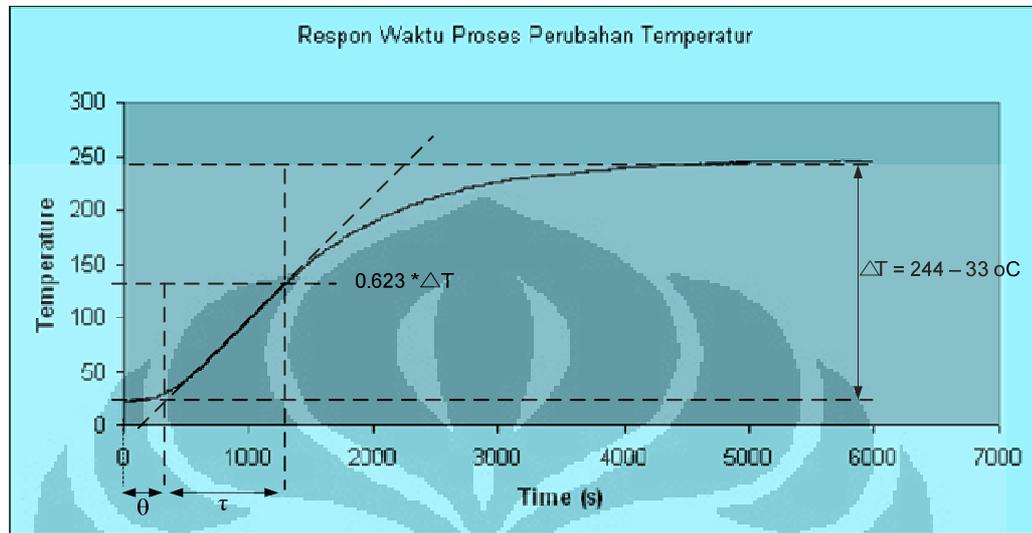
4.6 Pengujian Sistem PID Dengan Metode *Direct Synthesis*

Pengujian sistem pengendali dengan metode *direct synthesis* berfungsi mencari fungsi transfer dengan mencoba sistem tersebut. Dari Grafik daya 50% pada pengujian termokopel merupakan langkah awal yang dilakukan untuk menentukan nilai matematik dari respon sistem, untuk mendapatkan fungsi transfernya harus mencari nilai θ dan nilai τ . Untuk mendapatkan nilai θ , yakni dengan menarik garis lurus pada saat temperatur pengukuran awal (temperatur mula – mula). Kemudian dari titik temperatur mula – mula tersebut (dengan melihat grafik pada gambar 4.6) ditarik garis lurus horisontal untuk mendapatkan titik temu dengan garis miring pada kenaikan temperatur linier (dengan melihat grafik pada gambar 4.6), kemudian pada titik temu tersebut ditarik garis vertikal hingga diketahui posisi garis vertikal tersebut pada sumbu x untuk mengetahui nilai waktu. Nilai θ adalah nilai waktu pada sumbu x tersebut. Sedangkan untuk Mendapatkan nilai τ didapat melalui perhitungan seperti dibawah ini :

$$\tau = 0.632 * \Delta T \quad (4.2)$$

Dimana ΔT adalah nilai selisih antara T_{max} dan T_{min} dimana T_{max} adalah batas temperatur yang dicapai saat menggunakan daya 50 % dan T_{min} adalah temperatur mula - mula. Sehingga dengan melakukan perhitungan sesuai persamaan (4.2) maka didapatkan nilai τ (tau), kemudian nilai τ yang berupa nilai temperatur tersebut di beri titik pada sumbu y setelah itu ditarik garis horizontal hingga menyentuh kurva respon temperatur pada Gambar 4.6, kemudian setelah mengenai kurva tersebut diberi titik dan ditarik garis vertikal menuju sumbu x

untuk mengetahui nilai waktunya, maka didapat nilai waktu τ , tetapi nilai τ sesungguhnya adalah selisih waktu dari nilai τ yang didapat dengan nilai waktu θ .



Gambar 4.6 Mencari Nilai θ dan Nilai τ

Bila dilihat dari gambar diatas, dapat diketahui fungsi transfer dari system lalu dengan menggunakan metode *direct synthesis* dapat diperoleh nilai K_p , T_i dan T_d . Sebelum itu, dari gambar diatas diperoleh nilai $\theta = 115$ dan nilai $\tau = 1300 - 115 = 1085$. Untuk mendapatkan nilai ΔMV adalah $= 50 - 0 = 50\%$ dimana ΔMV ini adalah selisih nilai daya yang dipakai (50%) dengan nilai daya mula - mula (0%) dan $\Delta PV = ((244 - 32)/300)*100\% = (212/300)*100\% = 70,3\%$ dimana ΔPV adalah nilai prosentase temperatur pengukuran (244 °C dikurangi 32 °C) terhadap nilai temperatur maksimum yang diinginkan (300).

Dengan demikian persamaan yang didapat :

$$MV = K_p \left(E + \frac{1}{T_i} \int E dt + T_d \frac{dE}{dt} \right) \quad (4.3)$$

Dari persamaan diatas dapat mencari nilai K_p , T_i , dan T_d dengan menggunakan persamaan-persamaan dibawah ini, dimana ditentukan bahwa nilai $\tau C > \tau$ yaitu 1185:

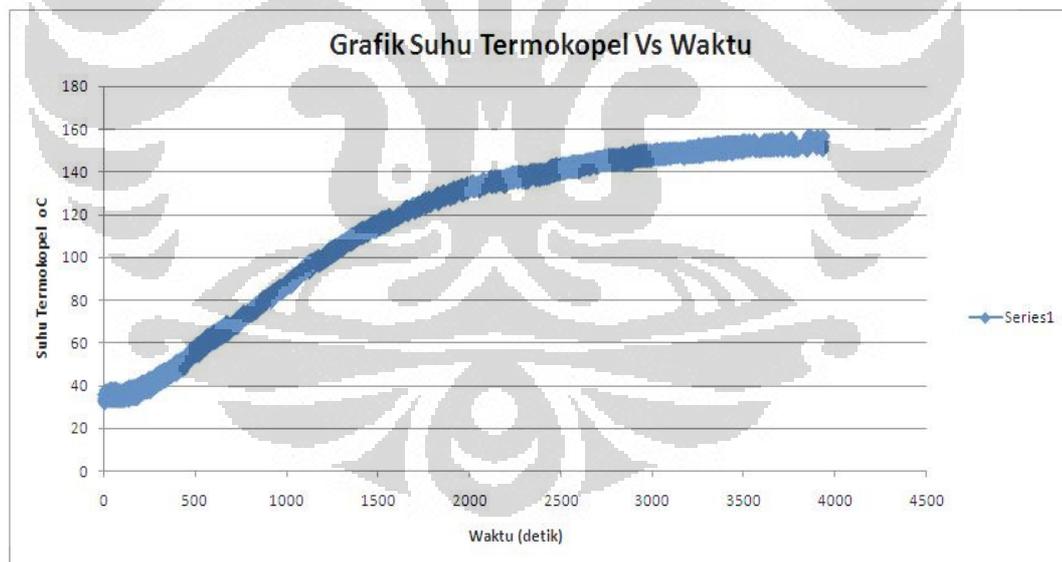
$$T_i = \frac{\theta}{2} + \tau = 57.5 + 1085 = 1142.5$$

$$T_d = \frac{\tau}{2\left(\frac{\tau}{\theta}\right)+1} = \frac{1085}{2(18.87)+1} = 54.61$$

$$K = \frac{\Delta PV}{\Delta MV} = \frac{\frac{244-32}{300} \times 100\%}{50\%} = \frac{70.3}{50} = 1.41$$

$$K_p = \frac{1}{K} \cdot \frac{2\left(\frac{\tau}{\theta}\right)+1}{2\left(\frac{\tau_c}{\theta}\right)+1} = \frac{1}{1.41} \cdot \frac{19.87}{21.61} = 0.65$$

Setelah itu, nilai dari persamaan-persamaan diatas yang telah didapatkan kemudian dimasukkan kedalam program pengendalian. Hasil uji sistem PID menghasilkan grafik suhu termokopel vs waktu dengan setpoint suhu pada heater 155 °C adalah sebagai berikut:



Gambar 4.7 Grafik hasil uji PID dengan setpoint 155 °C

BAB 5 KESIMPULAN DAN SARAN

5.1 Kesimpulan

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

- 1) Dengan *direct synthesis method* didapat persamaan pengendali berupa pengendali PID dengan nilai $K_p = 0.65$; $T_i = 1142.5$ dan $T_d = 54.61$
- 2) Persamaan temperatur yang di peroleh dan digunakan dalam program adalah $\text{Temperatur} = 0.808 \times \text{Nilai ADC} + 0.541$
- 3) Semakin Tinggi nilai PWM yang diberikan maka kecepatan putaran motor yang dihasilkan akan semakin tinggi.
- 4) Temperatur pada sasis mekanik heater hanya 45°C pada suhu aktual 240°C karena heater didesain menggunakan alas *glass wool* yang baik dalam menghambat panas sehingga panas dari heater dihambat oleh bahan tersebut, dengan demikian bagian mekanik pada box heater dinyatakan aman untuk disentuh.

5.2 Saran

- Perlu penelitian lebih lanjut untuk pengendalian PID pada motor agar respon kecepatan motor lebih baik.
- Sebaiknya tidak menyentuh resistor variabel pengatur tegangan penguatan pada Op-Amp, karena dapat berubah karena tergeser.

DAFTAR PUSTAKA

- [1] <http://www.omega.com/temperature/z/pdf/z021-032.pdf> (11/11/2011 15:00 WIB)
- [2] <http://product-image.tradeindia.com/00327595/b/0/RTD-Thermocouple-Sensors.jpg> (11/11/2011 16:00 WIB)
- [3] <http://www.thermometricscorp.com/thermocouple.html> (11/11/2011 15:30 WIB)
- [4] <http://www.thermometricscorp.com/thermocouple.html> (11/11/2011 16:15 WIB)
- [5] http://www.solarbotics.net/starting/200111_dc motor/pix/dc_motor_pieces.gif (12/11/2011 08:30 WIB)
- [6] <http://hyperphysics.phy-astr.gsu.edu/hbase/magnetic/imgmag/dcmop.gif> (12/11/2011 09:10 WIB)
- [7] Hughes, Fredrick W 1990, Panduan OP-AMP, Jakarta: PT Elex Media Komputindo
- [8] Kilian. Modern Control Technology : Components and Systems. Delmar, Inc.
- [9] Eko,Putra Agfianto Eko 2003, Belajar Mikrokontroller, Jakarta: Gava Media
- [10] Yamidi, Muhammad 2007, ” Rancang Bangun Sistem Pemanas Terkendali ”, Dalam Laporan Tugas akhir, Depok: Depertemen Fisika, Universitas Indonesia.
- [11] Clayton, George. Operational Amplifiers. Erlangga, 2002.

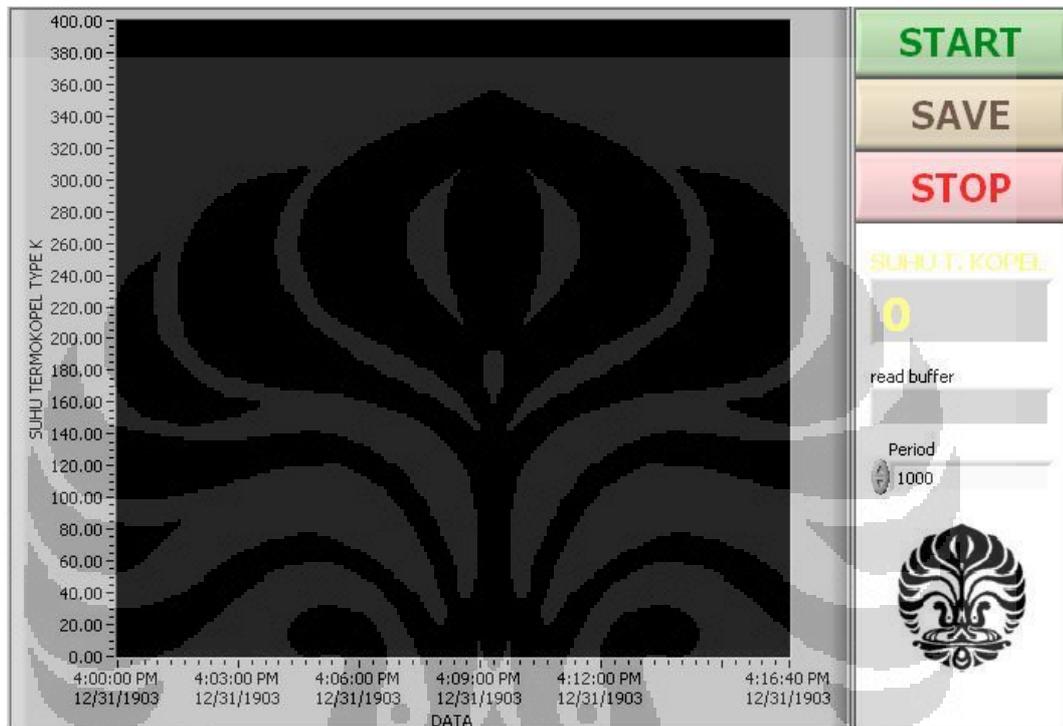
LAMPIRAN



DATA ADC TERMOKOPEL Vs DELTA TEMPERATUR

No	ADC Termokopel	Suhu Termometer	Nilai ADC T KOPEL - 71	Delta Temperatur
1	275	200	204	165
2	268	195	197	160
3	260	190	189	155
4	256	185	185	150
5	250	180	179	145
6	243	175	172	140
7	238	170	167	135
8	231	165	160	130
9	224	160	153	125
10	219	155	148	120
11	213	150	142	115
12	208	145	137	110
13	199	140	128	105
14	195	135	124	100
15	189	130	118	95
16	181	125	110	90
17	176	120	105	85
18	169	115	98	80
19	162	110	91	75
20	158	105	87	70
21	151	100	80	65
22	146	95	75	60
23	139	90	68	55
24	132	85	61	50
25	127	80	56	45
26	120	75	49	40
27	113	70	42	35
28	107	65	36	30
29	102	60	31	25
30	94	55	23	20
31	89	50	18	15
32	81	45	10	10
33	76	40	5	5
34	71	35	0	0

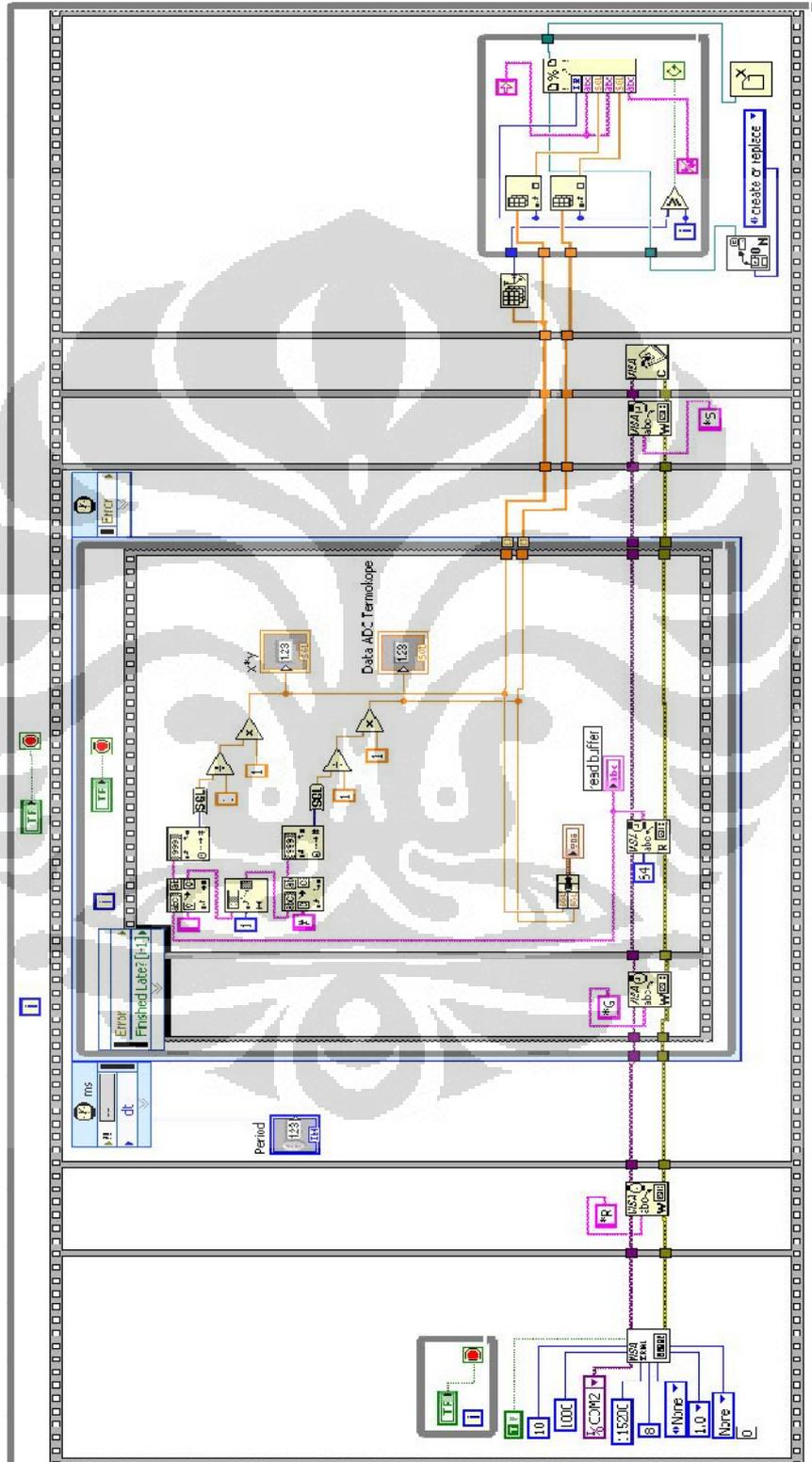
FRONT PANEL LABVIEW PENGAMBILAN DATA TEMPERATUR



FRONT PANEL LABVIEW PENGAMBILAN DATA ADC



DIAGRAM BLOK LABVIEW PENGAMBILAN DATA ADC



FRONT PANEL LABVIEW PEMANAS DAN PENGADUK PID

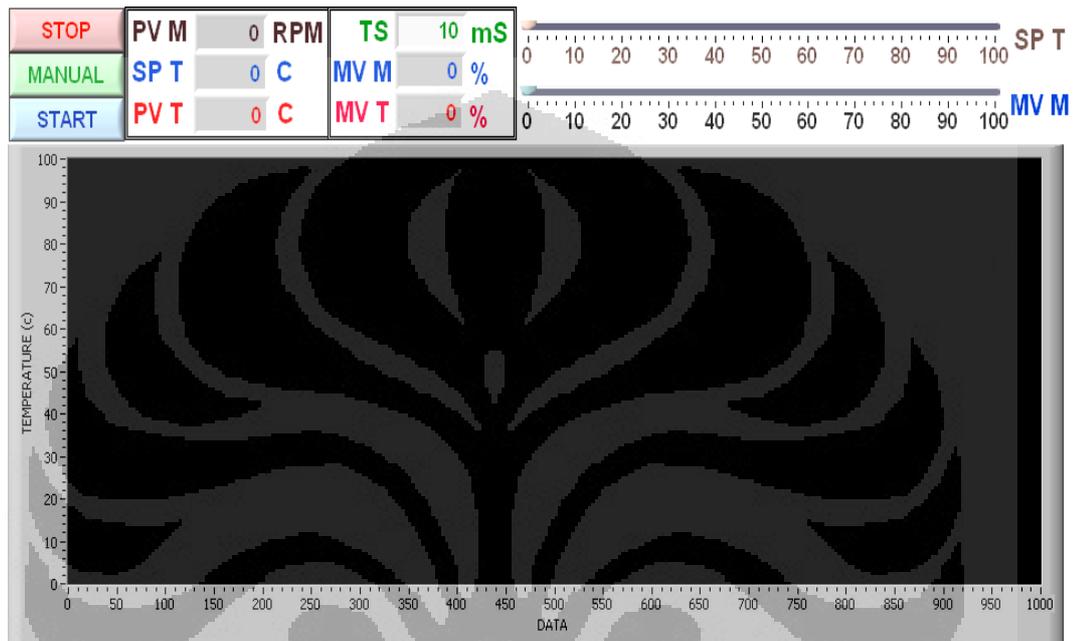
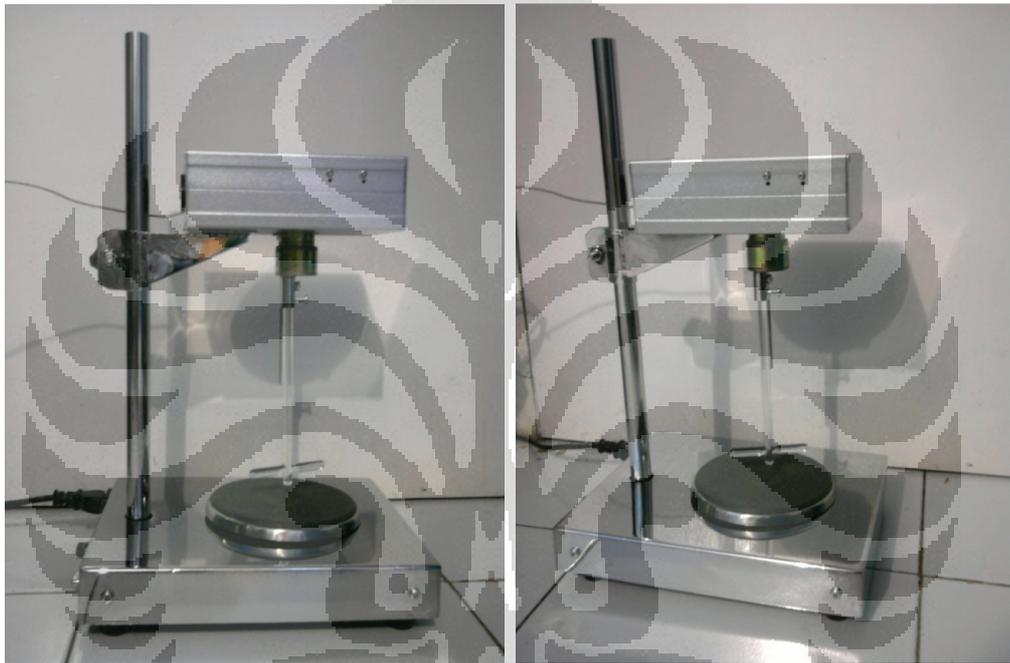


FOTO ALAT PEMANAS DAN PENGADUK TERINTEGRASI

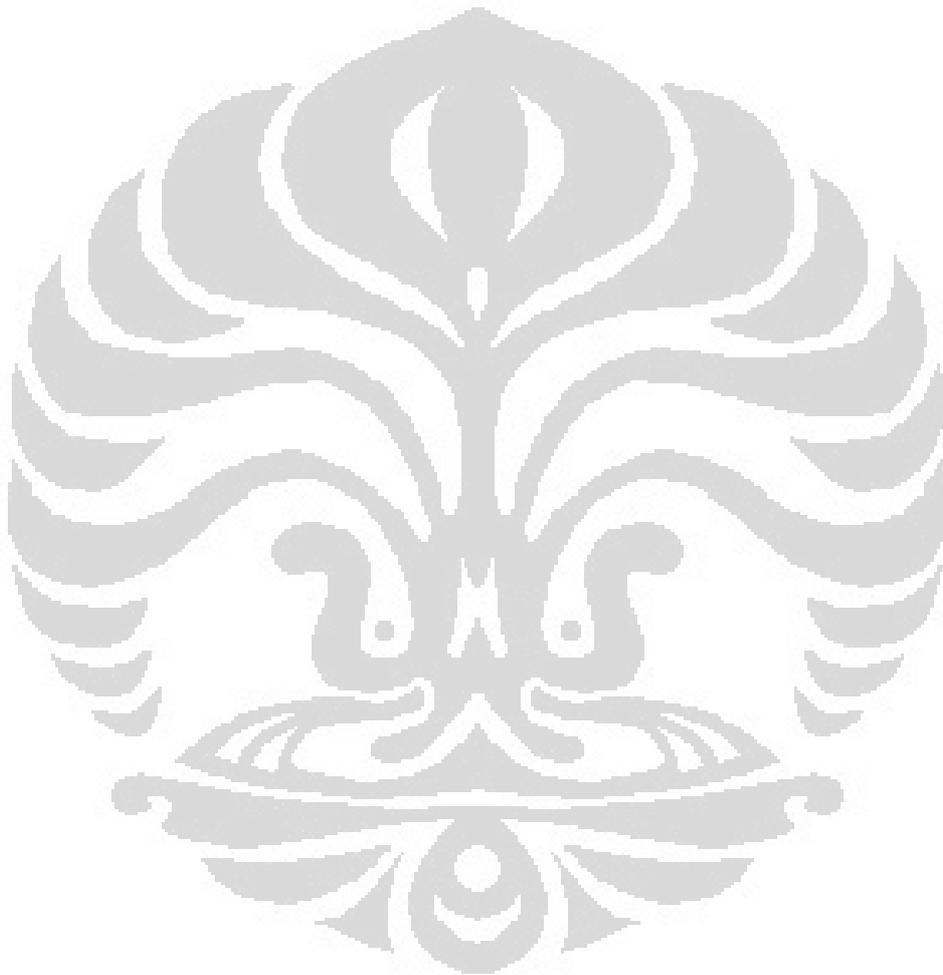


PROGRAM TEST NILAI KEYPAD

```
$regfile = "m16def.dat"
$crystal = 11059200
$baud = 115200
Config Lcdpin = Pin , Db4 = Porta.4 , Db5 = Porta.5 , Db6 = Porta.6 , Db7 =
Porta.7 , E = Porta.3 , Rs = Porta.2
Config Lcd = 20 * 4
Config Adc = Single , Prescaler = Auto , Reference = Avcc
Config Kbd = Portc , Debounce = 40
Dim Ulang As Byte
Dim Keypad As Byte

Main_program:
Cls
Locate 1 , 1
Lcd " TEST PROGRAM "
Locate 2 , 1
Lcd " KEYPAD MATRIKS "
Ulang = 1
Do
    Keypad = Getkbd()
    If Keypad < 16 Then
        Locate 2 , 1
        Lcd "      "
        Locate 2 , 1
        Lcd "TUTS VALUE = " ; Keypad
        Print "TUTS VALUE ON PRESS = " ; Keypad
```

```
    Waitms 200  
End If  
    Waitms 100  
Loop Until Ulang = 0  
Goto Main_program
```



PROGRAM MENU KEYPAD

```
$regfile = "m16def.dat"

$crystal = 11059200

$baud = 115200

On Urxc Data_serial

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

Config Lcd = 20 * 4

Config Kbd = Portc , Debounce = 60

'-----

Declare Sub Baca_menu_keypad()
Declare Sub Baca_flag_aktif()
Declare Sub Baca_keypad_temperatur()
Declare Sub Setting_suhu()
Declare Sub Setting_kecepatan()
Declare Sub Baca_keypad_kecepatan()
Declare Sub Baca_keypad_kecepatan_2()
Declare Sub Setting_waktu()
Declare Sub Tampilan_utama()

'-----

Dim Sp_temperatur As Word
Dim Sp_kecepatan As Word
Dim Sp_waktu As Word
Dim Act_temperatur As Word
Dim Act_kecepatan As Word
Dim Act_waktu As Word

'-----
```

```
Dim Keypad As Byte
Dim Nilai_keypad As Byte
'-----

Dim Balik As Bit
Dim Ulang As Byte
'-----

Dim Flag_setting_suhu As Bit
Dim Flag_setting_kecepatan As Bit
Dim Flag_setting_waktu As Bit
'-----

Dim Inisial As Byte
Dim I As Byte
'-----

Dim Data_seri As String * 1
Dim Kirim_flag_temp As Bit
Dim Kirim_flag_kec As Bit
Dim Kirim_flag_wkt As Bit

$eprom
Data_controller:
Data 30 , 30 , 30 , 30 , 30 , 30

$data

Cls

Cursor Off Noblink

Locate 1 , 1

Lcd " KEYPAD CONTROLLER "

Waitms 15

Locate 2 , 1
```

Lcd "* By: ADE M WIJAYA *"

Waitms 15

Locate 3 , 1

Lcd " BETA VERSION "

Wait 1

Cls

Locate 1 , 1

Lcd " DATA AKTUAL "

Locate 2 , 1

Lcd "TEMPERATUR ="

Locate 2 , 13

Lcd " "

Locate 2 , 19

Lcd Chr(223) ; "C"

Locate 3 , 1

Lcd "KECEPATAN ="

Locate 3 , 13

Lcd " "

Locate 3 , 18

Lcd "RPM"

Locate 4 , 1

Lcd "WAKTU ="

Locate 4 , 13

Lcd " "

Locate 4 , 18

Lcd "MNT"

Wait 1

'-----

Main_program:

Gosub Baca_menu_keypad

If Flag_setting_suhu = 1 Then

 Flag_setting_suhu = 0

 Gosub Setting_suhu

 Gosub Tampilan_utama

End If

If Flag_setting_kecepatan = 1 Then

 Flag_setting_kecepatan = 0

 Gosub Setting_kecepatan

 Gosub Tampilan_utama

End If

If Flag_setting_waktu = 1 Then

 Flag_setting_waktu = 0

 Gosub Setting_waktu

 Gosub Tampilan_utama

End If

'Call Kirim_data_serial

Goto Main_program

End

'end program

Data_serial:

Disable Interrupts

Data_seri = Inkey()

If Data_seri = "*" Then

```

Data_seri = Waitkey()
If Data_seri = "T" Then Kirim_flag_temp = 1
If Data_seri = "K" Then Kirim_flag_kec = 1
If Data_seri = "W" Then Kirim_flag_wkt = 1
End If
Enable Interrupts
Return
'-----
'-----
Sub Tampilan_utama()
  Readeeprom Sp_temperatur , Data_controller
  Readeeprom Sp_kecepatan
  Readeeprom Sp_waktu
  Readeeprom Act_temperatur
  Readeeprom Act_kecepatan
  Readeeprom Act_waktu
  Cls
  Locate 1 , 1
  Lcd "   MONITOR   "
  Locate 2 , 1
  Lcd "SP="
  Locate 2 , 4
  Lcd "  "
  Locate 2 , 4
  Lcd Sp_temperatur
  Locate 2 , 7
  Lcd "  "

```

Locate 2 , 9

Lcd "ACT="

Locate 2 , 13

Lcd " "

Locate 2 , 13

Lcd " "

Locate 2 , 18

Lcd Chr(223) ; "C"

Locate 3 , 1

Lcd "SP="

Locate 3 , 4

Lcd " "

Locate 3 , 8

Lcd Sp_kecepatan

Locate 3 , 7

Lcd " "

Locate 3 , 12

Lcd "ACT="

Locate 2 , 16

Lcd " "

Locate 2 , 16

Lcd " "

Locate 2 , 18

Lcd "RPM"

Locate 4 , 1

Lcd "SP="

```

Locate 4 , 4
Lcd " "
Locate 4 , 8
Lcd Sp_waktu
Locate 4 , 7
Lcd " "
Locate 4 , 12
Lcd "ACT="
Locate 4 , 16
Lcd " "
Locate 4 , 16
Lcd Act_waktu
Locate 2 , 18
Lcd "MNT"
End Sub
'-----
Sub Baca_menu_keypad()
Balik = 1
Do
  Keypad = Getkbd()
  Waitms 170
  If Keypad < 16 Then
    Select Case Keypad
      Case 3
        ' huruf A pada keypad ditekan
        Flag_setting_suhu = 1
      Case 7
        ' huruf B pada keypad ditekan
        Flag_setting_kecepatan = 1
    End Select
  End If
Loop

```

Case 11

' huruf C pada keypad ditekan

```
    Flag_setting_waktu = 1
End Select
Ulang = 1
Do
    Keypad = Getkbd()
    If Keypad = 16 Then
        Ulang = 0
    End If
Loop Until Ulang = 0
End If
Balik = 0
Loop Until Balik = 0
End Sub
'-----
Reset_1:
Sub Setting_suhu()
Readeeprom Sp_temperatur , Data_controller
Readeeprom Sp_kecepatan
Readeeprom Sp_waktu
Readeeprom Act_temperatur
Readeeprom Act_kecepatan
Readeeprom Act_waktu
Cursor Off Noblink
Cls
Locate 1 , 1
Lcd "  TEMPERATUR  "
```

```

Locate 2 , 1
Lcd "SET TEMP "           '1 - 10
Locate 2 , 10
Lcd "="
Locate 2 , 11
Lcd "  "
Locate 2 , 11
Lcd Sp_temperatur         '11 - 15
Locate 2 , 19
Lcd Chr(223) ; "C"       '19 - 20
Locate 3 , 1
Lcd "ACT TEMP ="
Locate 3 , 11
Lcd "  "
Locate 3 , 19
Lcd Chr(223) ; "C"
Gosub Baca_keypad_temperatur
Cursor Off Noblink
End Sub
'-----

'keypad A = 3
'keypad B = 7
'keypad C = 11
'keypad D = 15
'-----

Sub Baca_keypad_temperatur()
  Inisial = 0

```

Do

Cursor Off Noblink

Keypad = Getkbd()

Waitms 170

If Keypad < 16 Then

 If Keypad <> 3 Then

 If Keypad <> 7 Then

 If Keypad <> 11 Then

 If Keypad <> 15 Then

 If Keypad <> 15 Then

 'Ulang = 1

 'Do

 'Keypad = Getkbd()

 'If Keypad = 16 Then

 ' Ulang = 0

 'End If

 'Loop Until Ulang = 0

 Select Case Keypad

 Case 0

 Nilai_keypad = 1

 Case 1

 Nilai_keypad = 2

 Case 2

 Nilai_keypad = 3

 Case 4

 Nilai_keypad = 4

Case 5

Nilai_keypad = 5

Case 6

Nilai_keypad = 6

Case 8

Nilai_keypad = 7

Case 9

Nilai_keypad = 8

Case 10

Nilai_keypad = 9

Case 1

Nilai_keypad = 0

Case 13

Nilai_keypad = 0

Case 14

Nilai_keypad = 14

End Select

If Nilai_keypad < 10 Then

Inisial = Inisial + 1

If Inisial = 1 Then

Sp_temperatur = Nilai_keypad

Locate 2 , 11

Lcd " "

Locate 2 , 11

Cursor On Blink

Lcd Sp_temperatur

```

Else
    Sp_temperatur = 10 * Sp_temperatur
    Sp_temperatur = Sp_temperatur + Nilai_keypad
    Locate 2 , 11
    Lcd "  "
    Locate 2 , 11
    Cursor On Blink
    Lcd Sp_temperatur
End If
If Sp_temperatur > 250 Then Sp_temperatur = 250
End If
If Keypad = 14 Then
    Locate 2 , 11
    Lcd "  "
    Locate 2 , 11
    Lcd Sp_temperatur
    Writeeprom Sp_temperatur , Data_controller
    Writeeprom Sp_kecepatan
    Writeeprom Sp_waktu
    Writeeprom Act_temperatur
    Writeeprom Act_kecepatan

```

```

Writeeprom Act_waktu
Locate 4 , 6
Lcd "TERSIMPAN"
Waitms 700
End If
Loop Until Keypad = 14
End Sub
'-----
'-----
Reset_2:
Sub Setting_kecepatan()
Readeprom Sp_temperatur , Data_controller
Readeprom Sp_kecepatan
Readeprom Sp_waktu
Readeprom Act_temperatur
Readeprom Act_kecepatan
Readeprom Act_waktu
Cursor Off Noblink
Cls
Locate 1 , 1
Lcd "  KECEPATAN  "
Locate 2 , 1
Lcd "SET KEC "           '1 - 10
Locate 2 , 9
Lcd "= "
Locate 2 , 11
Lcd "  "

```

```

Locate 2 , 11
Lcd Sp_kecepatan                '11 - 15
Locate 2 , 16
Lcd "Rpm"                       '19 - 20
Locate 3 , 1
Lcd "ACT KEC = "
Locate 3 , 11
Lcd "  "
Locate 3 , 16
Lcd "Rpm"
Gosub Baca_keypad_kecepatan
Cursor Off Noblink
End Sub
'-----
Sub Baca_keypad_kecepatan()
  Inisial = 0
  Do
    Cursor Off Noblink
    Keypad = Getkbd()
    Waitms 170
    If Keypad < 16 Then
      If Keypad <> 3 Then
        If Keypad <> 7 Then
          If Keypad <> 11 Then
            If Keypad <> 15 Then
              If Keypad <> 15 Then
                Select Case Keypad

```

Case 0

 Nilai_keypad = 1

Case 1

 Nilai_keypad = 2

Case 2

 Nilai_keypad = 3

Case 4

 Nilai_keypad = 4

Case 5

 Nilai_keypad = 5

Case 6

 Nilai_keypad = 6

Case 8

 Nilai_keypad = 7

Case 9

 Nilai_keypad = 8

Case 10

 Nilai_keypad = 9

Case 1

 Nilai_keypad = 0

Case 13

 Nilai_keypad = 0

Case 14

 Nilai_keypad = 14

End Select

If Nilai_keypad < 10 Then

```

    Inisial = Inisial + 1
    If Inisial = 1 Then
        Sp_kecepatan = Nilai_keypad
        Locate 2 , 11
        Lcd "  "
        Locate 2 , 11
        Cursor On Blink
        Lcd Sp_kecepatan

    Else
        Sp_kecepatan = 10 * Sp_kecepatan
        Sp_kecepatan = Sp_kecepatan + Nilai_keypad
        Locate 2 , 11
        Lcd "  "
        Locate 2 , 11
        Cursor On Blink
        Lcd Sp_kecepatan
    End If

    If Sp_kecepatan > 300 Then Sp_kecepatan = 300
    End If

```

```

If Keypad = 14 Then
    Locate 2 , 11
    Lcd "  "
    Locate 2 , 11
    Lcd Sp_kecepatan
    Writeeprom Sp_temperatur , Data_controller
    Writeeprom Sp_kecepatan
    Writeeprom Sp_waktu
    Writeeprom Act_temperatur
    Writeeprom Act_kecepatan
    Writeeprom Act_waktu
    Locate 4 , 6
    Lcd "TERSIMPAN"
End If
Loop Until Keypad = 14
End Sub
'-----
Sub Setting_waktu()
Readeeprom Sp_temperatur , Data_controller
Readeeprom Sp_kecepatan
Readeeprom Sp_waktu
Readeeprom Act_temperatur
Readeeprom Act_kecepatan
Readeeprom Act_waktu
Cursor Off Noblink
Cls
Locate 1 , 1

```

```
Lcd " WAKTU "
```

Locate 2 , 1

```
Lcd "SET WKT " '1 - 10
```

Locate 2 , 9

```
Lcd "= "
```

Locate 2 , 11

```
Lcd " "
```

Locate 2 , 11

```
Lcd Sp_waktu '11 - 15
```

Locate 2 , 16

```
Lcd "MNT" '19 - 20
```

Locate 3 , 1

```
Lcd "ACT WKT = "
```

Locate 3 , 11

```
Lcd Act_waktu
```

Locate 3 , 16

```
Lcd "MNT"
```

```
'Gosub Baca_keypad_waktu
```

```
Cursor Off Noblink
```

```
End Sub
```

PROGRAM PENGADUK DAN PEMANAS UNTUK DAYA 50%

\$regfile = "m8def.dat"

\$crystal = 11059200

\$baud = 115200

'-----'

Enable Interrupts

On Ovfo Switching_program

On Urxc Dataseri_in

Enable Ovfo

Enable Urxc

'====='

Dim Periode_1ms As Single

Dim Periode_10ms As Byte

Dim Periode_10ms_motor As Byte

Dim Periode_1s As Byte

Dim Mv_var_heater As Byte

Dim Mv_var_motor As Byte

Dim Power_heater As Byte

Dim Power_motor As Byte

Dim Periode_1s_flag As Bit

Dim Periode_10ms_motor_flag As Bit

Dim Ulang As Bit

Dim Start_program_flag As Bit

Dim Stop_program_flag As Bit

Dim Kirim_data_flag As Bit

Dim Data_seri As Byte

Dim Adc_lm35 As Word

Dim Adc_t_kopel As Word

Dim Suhu_lm35 As Single

Dim Suhu_lm35_round As Single

Dim Suhu_t_kopel As Single

Dim Suhu_t_kopel_round As Single

Dim Suhu_lm35_int As Integer

Dim Suhu_t_kopel_int As Integer

'=====

Config Timer0 = Timer , Prescale = 8

'Config Timer1 = Pwm , Pwm = 10 , Compare A Pwm = Clear Up , Compare B
Pwm = Clear Up , Prescale = 256

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

Config Portb.0 = Output 'Heater

Config Portb.1 = Output 'Motor

Config Portb.2 = Output

'=====

Heater Alias Portb.0

Motor Alias Portb.1

Main_program:

Start_program_flag = 0

Kirim_data_flag = 0

Ulang = 1

Start Adc

Start Timer0

Do

```

Mv_var_heater = 50
Mv_var_motor = 50
If Start_program_flag = 1 Then
  Adc_lm35 = Getadc(3)
  Adc_t_kopel = Getadc(4)
  Suhu_lm35 = Adc_lm35 * 4.9
  Suhu_lm35 = Suhu_lm35 / 10
  Suhu_lm35 = Round(suhu_lm35)
  Suhu_lm35_round = Suhu_lm35
  Suhu_t_kopel = Adc_t_kopel - 71
  Suhu_t_kopel = Suhu_t_kopel * 0.808
  Suhu_t_kopel = Suhu_t_kopel + 0.541
  Suhu_t_kopel = Suhu_t_kopel + Suhu_lm35_round
  Suhu_t_kopel_round = Round(suhu_t_kopel)
  If Kirim_data_flag = 1 Then
    Kirim_data_flag = 0
    Print Suhu_lm35_round ; ":" ; Suhu_t_kopel_round ; "#"
  End If
End If
Loop Until Ulang = 0
Goto Main_program
End

'=====
'++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Switching_program:
Periode_10ms = Periode_10ms + 1
If Periode_10ms = 54 Then

```

```
Periode_10ms = 0
Periode_1s = Periode_1s + 1
If Power_heater = 0 Then
    Reset Heater
Else
    Power_heater = Power_heater - 1
    Set Heater
End If
If Periode_1s = 100 Then
    Periode_1s_flag = 1
    Periode_1s = 0
    Power_heater = Mv_var_heater
End If
End If
Periode_1ms = Periode_1ms + 1
If Periode_1ms = 5 Then
    Periode_1ms = 0
    Periode_10ms_motor = Periode_10ms_motor + 1
    If Power_motor = 0 Then
        Reset Motor
    Else
        Power_motor = Power_motor - 1
        Set Motor
    End If
    If Periode_10ms_motor = 100 Then
        Periode_10ms_motor_flag = 1
        Periode_10ms_motor = 0
```

```
    Power_motor = Power_motor - 1
    Power_motor = Mv_var_motor
End If
End If
Return
=====
-----
Dataseri_in:
Disable Interrupts
Data_seri = Inkey()
If Data_seri = "*" Then
    Data_seri = Waitkey()
    If Data_seri = "R" Then Start_program_flag = 1
    If Data_seri = "G" Then Kirim_data_flag = 1
    If Data_seri = "S" Then Ulang = 0
End If
Enable Interrupts
Return
```

PROGRAM KONTROL HEATER DAN KECEPATAN

\$regfile = "m8def.dat"

\$crystal = 11059200

\$baud = 19200

'-----

On Ovf2 Switching_program

On Urxc Dataseri_in

Enable Interrupts

Enable Ovf2

Enable Urxc

'=====

Dim Pid_heater_flag As Bit

Dim Manual_flag As Bit

Dim Ulang As Bit

Dim Start_prog_flag As Bit

Dim Kirim_data_flag As Bit

Dim Periode_10_ms As Byte

Dim Periode_80_ms As Byte

Dim Periode_2_s As Byte

Dim Cacahan As Byte

Dim Mv_heater As Integer

Dim Pwr_heater As Integer

Dim Mv_motor As Integer

Dim Spd_motor As Integer

Dim Data_adc As Word

Dim Pv_temp As Single

```

Dim Sp_temp As Single
Dim Error As Single
Dim Error_old As Single
Dim Delta_error As Single
Dim Sigma_error As Single
Dim Mv As Single
Dim Mv_p As Single
Dim Mv_i As Single
Dim Mv_d As Single
Dim Batas_atas As Single
Dim Batas_bawah As Single
Dim Data_converter As Single
Dim Data_converter1 As Single
Dim Data_seris As String * 1
Dim Pwr_heater_str As String * 5
Dim Pwr_motor_str As String * 5
Dim Sp_temp_str As String * 5
Dim Temp_str As String * 5
'=====
Config Timer0 = Counter , Edge = Falling
Config Timer1 = Pwm , Pwm = 10 , Compare A Pwm = Clear Up , Compare B
Pwm = Clear Up , Prescale = 8
Config Timer2 = Timer , Prescale = 8
Config Adc = Single , Prescaler = Auto , Reference = Avcc
Config Portb.0 = Output           'Heater
Config Portb.1 = Output           'Motor
Config Portb.2 = Output

```

Heater Alias Portb.0

'=====

```
Const Kp = 1
Const Ti = 0.1
Const Td = 0.01
Start_prog_flag = 0
Do
If Start_prog_flag = 1 Then
    Start Adc
    Start Timer0
    Start Timer1
    Start Timer2
    Batas_atas = 50 * Ti
    Batas_bawah = -1 * Batas_atas
    If Manual_flag = 1 Then
        Data_converter = Val(pwr_heater_str)
        Data_converter = Data_converter / 10
        Data_converter = Round(data_converter)
        Mv_heater = Data_converter
        Data_converter = Val(pwr_motor_str)
        Data_converter = 10.23 * Data_converter
        Data_converter = Round(data_converter)
        Mv_motor = Data_converter
        Data_adc = Getadc(3)
        Data_converter = Data_adc / 2
        Data_adc = Getadc(4)
        Data_adc = Data_adc - 71
```

```
Data_converter1 = 0.808 * Data_adc
Data_converter1 = Data_converter1 + 0.541
Data_converter1 = Data_converter1 + Data_converter
Data_converter1 = Data_converter1 / 3
Temp_str = Fusing(data_converter1 , "#.#")
Data_converter1 = Round(data_converter1)
Pwm1a = Mv_motor
Pwm1b = Mv_motor
Data_converter = Cacahan / 230
Data_converter = Data_converter * 480
Data_converter = Round(data_converter)
Spd_motor = Data_converter
Sp_temp_str = "0"
Else
Data_converter = Val(sp_temp_str)
Data_converter = Data_converter / 10
Sp_temp = Data_converter
Data_converter = Val(pwr_motor_str)
Data_converter = 10.23 * Data_converter
Data_converter = Round(data_converter)
Mv_motor = Data_converter
Data_adc = Getadc(3)
Data_converter = Data_adc / 2
Data_adc = Getadc(4)
Data_adc = Data_adc - 71
Data_converter1 = 0.808 * Data_adc
Data_converter1 = Data_converter1 + 0.541
```



```

Data_converter1 = Data_converter1 + Data_converter
Data_converter1 = Data_converter1 / 3
Temp_str = Fusing(data_converter1 , "#.#")
Pv_temp = Data_converter1
Pwm1a = Mv_motor
Pwm1b = Mv_motor
Data_converter = Cacahan / 230
Data_converter = Data_converter * 480
Data_converter = Round(data_converter)
Spd_motor = Data_converter
Error = Sp_temp - Pv_temp
Delta_error = Error - Error_old
Error_old = Error
Mv_p = Kp * Error
Sigma_error = Sigma_error + Mv_p
If Sigma_error > Batas_atas Then Sigma_error = Batas_atas
If Sigma_error < Batas_bawah Then Sigma_error = Batas_bawah
Mv_i = Sigma_error * 2
Mv_i = Mv_i / Ti
Mv_d = Kp * Delta_error
Mv_d = Mv_d * Td
Mv_d = Mv_d / 2
Mv = Mv_p + Mv_i
Mv = Mv + Mv_d
If Mv > 100 Then Mv = 100
If Mv < 0 Then Mv = 0
Mv = Round(mv)

```

```

    Mv_heater = Mv
End If
If Kirim_data_flag = 1 Then
    Kirim_data_flag = 0
    Print Spd_motor ; ":" ; Sp_temp_str ; ":" ; Mv_heater ; ":" ; Temp_str ; "#"
End If
Else
    Stop Adc
    Stop Timer0
    Stop Timer1
    Stop Timer2
    Reset Heater
End If
Loop
'=====
'++++++

Switching_program:
If Periode_10_ms = 54 Then
    Periode_10_ms = 0
    If Periode_80_ms = 8 Then
        Periode_80_ms = 0
        Cacahan = Counter0
        Counter0 = 0
        Start Timer0
    Else
        Periode_80_ms = Periode_80_ms + 1
    
```

```

End If
If Periode_2_s = 200 Then
    Periode_2_s = 0
    Pwr_heater = 2 * Mv_heater
    Pid_heater_flag = 1
Else
    Periode_2_s = Periode_2_s + 1
    If Pwr_heater = 0 Then
        Reset Heater
    Else
        Pwr_heater = Pwr_heater - 1
        Set Heater
    End If
End If
Else
    Periode_10_ms = Periode_10_ms + 1
End If
Return
=====
-----

Dataseri_in:
Disable Interrupts
Data_seri = Inkey()
If Data_seri = "*" Then
    Data_seri = Waitkey()
    If Data_seri = "M" Then
        Kirim_data_flag = 1

```

```
Manual_flag = 1
Data_seri = Waitkey()
Ulang = 1
Pwr_heater_str = ""
Do
    Data_seri = Waitkey()
    If Data_seri = ":" Then
        Ulang = 0
    Else
        Pwr_heater_str = Pwr_heater_str + Data_seri
    End If
Loop Until Ulang = 0
Ulang = 1
Pwr_motor_str = ""
Do
    Data_seri = Waitkey()
    If Data_seri = "#" Then
        Ulang = 0
    Else
        Pwr_motor_str = Pwr_motor_str + Data_seri
    End If
Loop Until Ulang = 0
End If
If Data_seri = "A" Then
    Manual_flag = 0
    Kirim_data_flag = 1
    Data_seri = Waitkey()
```

```
Ulang = 1
Sp_temp_str = ""
Do
    Data_seri = Waitkey()
    If Data_seri = ":" Then
        Ulang = 0
    Else
        Sp_temp_str = Sp_temp_str + Data_seri
    End If
Loop Until Ulang = 0
Ulang = 1
Pwr_motor_str = ""
Do
    Data_seri = Waitkey()
    If Data_seri = "#" Then
        Ulang = 0
    Else
        Pwr_motor_str = Pwr_motor_str + Data_seri
    End If
Loop Until Ulang = 0
End If
If Data_seri = "R" Then Start_prog_flag = 1
If Data_seri = "S" Then Start_prog_flag = 0
End If
Enable Interrupts
Return
```