



TUGAS AKHIR TF 145565

**RANCANG BANGUN SISTEM PENGENDALIAN
TEKANAN PADA ALIRAN UAP**

**Azania Arnada Auludyah
NRP. 2414 031 012**

**Dosen Pembimbing
Totok Ruki Biyanto, ST, MT, Ph.D
NIP. 19710702 199802 1 001**

**DEPARTEMEN TEKNIK INSTRUMENTASI
FAKULTAS VOKASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2017**



TUGAS AKHIR – TF 145565

**RANCANG BANGUN SISTEM PENGENDALIAN
TEKANAN PADA ALIRAN UAP**

**Azania Arnada Auludyah
NRP. 2414 031 012**

**Dosen Pembimbing
Totok Ruki Biyanto, ST, MT, Ph.D
NIP. 19710702 199802 1 001**

**DEPARTEMEN TEKNIK INSTRUMENTASI
FAKULTAS VOKASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2017**



FINAL PROJECT – TF 145565

***DESIGN OF PRESSURE CONTROL SYSTEMS ON
STEAM FLOW***

**Azania Arnada Auludyah
NRP. 2414 031 012**

Advisor Lecturer
Totok Ruki Biyanto, ST, MT, Ph.D
NIP. 19710702 199802 1 001

***DEPARTMENT OF INSTRUMENTATION ENGINEERING
FACULTY OF VOCATIONAL
SEPULUH NOMPENBER INSTITUTE OF TECHNOLOGY
SURABAYA 2017***

RANCANG BANGUN SISTEM PENGENDALIAN TEKANAN PADA ALIRAN UAP

TUGAS AKHIR

Oleh:

Azania Arnada Auludyah
NRP. 2414 031 012

Surabaya, 24 Juli 2017
Mengetahui / Menyetujui

Dosen Pembimbing



Totok Ruki Biyanto, ST, MT, Ph.D
NIP. 19710702 199802 1 001

Kepala Departemen
Teknik Instrumentasi



RANCANG BANGUN SISTEM PENGENDALIAN TEKANAN PADA ALIRAN UAP

TUGAS AKHIR

Diajukan untuk Memenuhi Salah Satu Sarat
Memperoleh Gelar Ahli Madya
pada
Program Studi D3 Teknik Instrumentasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember

Oleh:

Azania Arnada Auludyah
NRP. 2414 031 012

Disetujui oleh Tim Penguji Tugas Akhir :

1. Totok Ruki Biyanto, ST, MT, Ph.D..........(Dosen Pembimbing I)
2. Detak Yan Pratama ST, M.Sc..........(Ketua Tim Penguji)
3. Gunawan Nugroho, ST, MT, Ph.D.....(Dosen Penguji I)

**SURABAYA
JULI 2017**

RANCANG BANGUN SISTEM PENGENDALIAN TEKANAN PADA ALIRAN UAP

Nama Mahasiswa : Azania Arnada Auludyah
NRP : 2414 031 012
Departemen : Teknik Instrumentasi Vokasi-ITS
Dosen Pembimbing : Totok Ruki Biyanto, ST, MT, Ph.D

Abstrak

Ketel uap merupakan peralatan yang banyak digunakan pada industri-industri untuk pembangkit tenaga. Pada sebuah industri, ketel uap mempunyai peranan penting yaitu sebagai sumber penghasil uap dimana uap tersebut berguna untuk menggerakkan turbin uap yang menghasilkan tenaga untuk menggerakkan sebagian besar peralatan pada industri. Energi dalam (*internal energy*) dari air akan meningkat seiring dengan meningkatnya temperatur dan tekanan. Pengukuran tekanan udara memegang peranan sangat penting dalam bidang industri. Saat ini banyak industri yang memanfaatkan konsep tekanan dalam proses industri, sehingga pengukuran diperlukan dalam pemantauan dan pengendalian suatu proses. Sehingga dalam tugas akhir ini perlu adanya sebuah rancang bangun sistem pengendalian tekanan pada aliran uap. Sistem ini menggunakan sensor MPX5050GP yang memiliki sensitivitas sebesar 93.238488 mV/kPa. Didapatkan juga nilai *error steady state* sebesar 0.04 dan *maximum overshoot* sebesar 1.8%. Pada pengendalian tekanan ini tidak memiliki *settling time* karena sistem bekerja secara *on off*.

Kata Kunci: *Steam*, Pengendalian Tekanan, Labu Pemanas, Sensor MPX5050GP

DESIGN OF PRESSURE CONTROL SYSTEMS ON STEAM FLOW

Name : Azania Arnada Auludyah
NRP : 2414 031 012
Department : *Instrumentation Engineering*
Vokasi-ITS
Supervisor : Totok Ruki Biyanto, ST, MT, Ph.D

Abstract

Steam boilers are equipment that is widely used in industries for power generation. In an industry, the steam boiler has an important role as a source of steam-producing where steam is useful for moving steam turbines that generate power to drive most of the equipment in the industry. The internal energy of the air will increase with temperature and pressure. Measurement of air pressure plays a very important role in the field of industry. Today many industries are undergoing a process. In this final project, it is necessary to design of pressure control system on steam flow. This system uses the MPX5050GP sensor which has a sensitivity of 93.238488 mV / kPa. Also found steady state error value of 0.04, maximum overshoot of 1.8%. At this pressure control does 'nt have settling time because the system is on off.

Keywords: *Steam, Pressure Control, Heat Pump, Seonsor MPX5050GP.*

KATA PENGANTAR

Puji syukur kehadirat Allah SWT atas rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan laporan Tugas Akhir yang berjudul **“RANCANG BANGUN SISTEM PENGENDALIAN TEKANAN PADA ALIRAN UAP”** dengan tepat waktu. terselesaikannya laporan ini juga tak luput dari dukungan dan peran dari orangtua dan keluarga besar serta berbagai pihak. Untuk itulah dalam kesempatan ini penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Bapak Dr. Ir. Purwadi Agus Darwito, M.Sc selaku Kepala Departemen Teknik Instrumentasi.
2. Bapak Totok Ruki Biyanto, ST, MT, Ph.D dan Dr. Gunawan Nugroho, ST, MT selaku pembimbing Tugas Akhir yang telah membina dengan baik dan sabar.
3. Ibu Ronny Noriyati, M. Kes selaku Dosen Wali penulis.
4. Kedua orang tua yang tidak henti-hentinya memberi semangat dan doa untuk menyelesaikan Tugas akhir ini.
5. Mas Adam selaku kakak yang selalu membantu adiknya dalam penyelesaian Tugas Akhir.
6. Saudari Heni Kustianingsih selaku sahabat penulis yang telah bersama-sama berjuang dalam pengerjaan Tugas Akhir ini hingga selesai.
7. Team Pak TRB, Empu, Nova, Uis, Kakan, dan Ferry yang senantiasa mendengar keluh kesah penulis.
8. Seluruh Asisten Laboratorium *Workshop Instrumentasi*, Laboratorium Instrumentasi dan Kontrol, Laboratorium Pengukuran Fisis, dan Sahabat Zelena yang telah membantu dalam pengerjaan Tugas Akhir penulis.
9. Pak Ali yang telah membantu dalam pembuatan labu pemanas.
10. Mas Fahmi dan Mas Alim yang telah membantu dalam pengkodean Tugas Akhir penulis.
11. H2 Cutting yang telah membantu dalam pengerjaan Tugas Akhir penulis.

12. Teman-teman Teknik Instrumentasi dan S1 Teknik Fisika angkatan 2014 FTI-ITS.
13. Serta semua pihak yang tidak dapat disebutkan satu persatu.

Penulis menyadari bahwa laporan ini masih kurang sempurna. Oleh karena itu penulis menerima segala masukan baik berupa saran, kritik, dan segala bentuk tegur sapa demi kesempurnaan laporan ini.

Demikian laporan Tugas Akhir ini penulis persembahkan dengan harapan dapat bermanfaat dalam akademik baik bagi penulis sendiri maupun bagi pembaca.

Surabaya, 13 Juni 2017

Penulis

DAFTAR ISI

	Hal
HALAMAN JUDUL	i
TITLE OF PAGE.....	ii
LEMBAR PENGESAHAN I	iii
LEMBAR PENGESAHAN II.....	iv
ABSTRAK	v
ABSTRACT	vi
KATA PENGANTAR.....	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	xi
DAFTAR TABEL	xii

BAB I PENDAHULUAN

1.1 Latar Belakang	1
1.2 Permasalahan.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	2
1.5 Manfaat	2
1.6 Sistematika Laporan	3

BAB II TINJAUAN PUSTAKA

2.1 <i>Steam</i>	5
2.2 Labu Didih	7
2.3 Sistem Pengendalian Otomatis	7
2.4 Mikrokontroler ATmega16.....	9
2.5 Motor Servo	10
2.6 Sensor Tekanan	12
2.7 <i>Water Flow Sensor</i>	13
2.8 LCD	14
2.9 Pengendalian <i>ON-OFF</i>	14
2.10 Karakteristik Statik	15
2.11 Teori Ketidakpastian.....	16

BAB III PERANCANGAN DAN PEMBUATAN ALAT	
3.1 Diagram Alir dan Diagram Blok Perancangan Alat	19
3.2 Gambaran Umum	21
3.3 Studi Literatur Alat Ukur Tekanan dan Metode Kontrol Tekanan.....	22
3.4 Perancangan Sistem dan Pembuatan Hardware serta Software Alat Pengendalian Tekanan.....	22
3.5 Integrasi <i>Hardware</i> dan <i>Controller</i>	24
3.6 Pengujian Sistem Pengukuran dan Sistem Pengendalian	24
3.7 Pengambilan dan Analisis Data.....	25
3.8 Prosedur Operasional.....	26
3.9 Penulisan Laporan	27
BAB IV PENGUJIAN DAN ANALISIS DATA	
4.1 Pengujian Sensor Tekanan MPX5050GP	29
4.2 Data Spesifikasi Alat	34
4.3 Pengujian Respon Sistem Kontrol Tekanan pada Labu Pemanas	41
4.4 Pembahasan.....	45
BAB V PENUTUP	
5.1 Kesimpulan	47
5.2 Saran	47
DAFTAR PUSTAKA	
LAMPIRAN A (<i>LISTING PROGRAM PADA CODE VISION AVR</i>)	
LAMPIRAN B (<i>DATA SHEET MINIMUM SYSTEM ATMEGA16</i>)	
LAMPIRAN C (<i>DATA SHEET MPX5050GP</i>)	
LAMPIRAN D (<i>DATA SHEET WATER FLOW SENSOR G1/2</i>)	

DAFTAR GAMBAR

	Hal
Gambar 2.1	Kurva <i>Steam</i> Jenuh..... 5
Gambar 2.2	Labu Didih..... 7
Gambar 2.3	Diagram Blok Sistem Pengendalian Otomatis 8
Gambar 2.4	Pin-pin ATmega16..... 9
Gambar 2.5	Motor Servo MG 996R Tower Pro 10
Gambar 2.6	Sensor Tekanan MPX5050GP 11
Gambar 2.7	Fisik dan Skematik <i>Water Flow Sensor G1/2</i> ... 12
Gambar 2.8	LCD <i>Character 4x20</i> 13
Gambar 2.9	Aksi Kontrol On Off 14
Gambar 2.10	Tabel <i>T-student</i> 17
Gambar 3.1	Diagram Alir Tugas Akhir..... 19
Gambar 3.2	Diagram Blok Sistem Pengendalian Tekanan... 20
Gambar 3.3	Desain <i>Plant</i> Labu Pemanas 21
Gambar 3.4	P&ID Sistem Pengendalian Tekanan pada Labu Pemanas..... 22
Gambar 3.5	<i>Software Code</i> Vision AVR..... 24
Gambar 4.1	<i>Mini Plant</i> Labu Pemanas..... 29
Gambar 4.2	Penempatan Sensor MPX5050GP..... 30
Gambar 4.3	Grafik Pengujian Sensitivitas Sensor MPX5050GP 31
Gambar 4.4	Grafik Histerisis Sensor MPX5050GP 35
Gambar 4.5	Grafik <i>Open Loop</i> Sistem Pengendalian Tekanan..... 42
Gambar 4.6	Diagram Blok Sistem Pengendalian Tekanan... 42
Gambar 4.7	Grafik <i>Close Loop</i> Sistem Pengendalian Tekanan..... 43
Gambar 4.8	Grafik Respon <i>Tracking Set Point</i> Sistem Pengendalian Tekanan 44

DAFTAR TABEL

	Hal
Tabel 4.1	hasil Pengujian V_{Out} pada Sensor MPX5050GP 30
Tabel 4.2	Data Pengujian Naik Sensor MPX5050GP (KPa) 32
Tabel 4.3	Data Pengujian Naik Sensor MPX5050GP (Psi) 32
Tabel 4.4	Data Pengujian Turun Sensor MPX5050GP (KPa) 33
Tabel 4.5	Data Pengujian Turun Sensor MPX5050GP (Psi) 33
Tabel 4.6	Pengambilan Data Naik dan Turun pada Sensor MPX5050GP 34
Tabel 4.7	Data Kalibrasi pada Sensor MPX5050GP (A)..... 37
Tabel 4.8	Data Kalibrasi pada Sensor MPX5050GP (B)..... 38
Tabel 4.8	Data Kalibrasi pada Sensor MPX5050GP (C)..... 38

BAB I

PENDAHULUAN

1.1 Latar Belakang

Ketel uap merupakan peralatan yang banyak digunakan pada industri-industri untuk pembangkit tenaga. Pada sebuah industri, ketel uap mempunyai peranan penting yaitu sebagai sumber penghasil uap dimana uap tersebut berguna untuk menggerakkan turbin uap yang menghasilkan tenaga untuk menggerakkan sebagian besar peralatan pada industri. Uap atau fluida panas yang dihasilkan kemudian disirkulasikan dari ketel uap untuk berbagai proses dalam aplikasi pemanasan. Ketel uap atau pembangkit uap adalah salah satu dari sekian banyak peralatan dalam siklus energi termal yang bertujuan untuk merubah air menjadi uap bertekanan sehingga dapat digunakan untuk mengoperasikan beberapa mesin yang menggunakan tenaga uap. Pada dasarnya, uap yang dihasilkan oleh ketel uap digunakan untuk membangkitkan tenaga mekanik atau mensuplai panas untuk keperluan industri (*manufacturing process*). Bentuk dari ketel uap secara garis besar merupakan suatu bejana tertutup, dimana kalor dari pembakaran bahan bakar dipindahkan ke air melalui ruang bakar dan bidang-bidang pemanasan.[1]

Energi dalam (*internal energy*) dari air akan meningkat seiring dengan meningkatnya temperatur dan tekanan. Pengukuran tekanan udara memegang peranan sangat penting dalam bidang industri. Saat ini banyak industri yang memanfaatkan konsep tekanan dalam proses industri, sehingga pengukuran diperlukan dalam pemantauan dan pengendalian suatu proses. Maka dari itu sistem pengontrolan tekanan dibuat selain dapat mendukung proses produksi juga untuk keamanan apabila terjadi *error* atau kelebihan *supply* bahan bakar yang mengakibatkan pemanasan yang berlebihan sehingga diperlukan pengendalian tekanan pada sistem tersebut. Untuk mendukung itu semua, maka diperlukan adanya otomatisasi yang dapat mengurangi tingkat kesalahan yang disebabkan oleh kesalahan manusia. Oleh karena itu dibuat tugas akhir ini dengan judul

“RANCANG BANGUN SISTEM PENGENDALIAN TEKANAN PADA ALIRAN UAP”.

1.2 Permasalahan

Berdasarkan latar belakang yang dijelaskan diatas, maka permasalahan dalam tugas akhir ini adalah:

- a. Perlu dirancang sistem pengendalian mulai dari sensor, *controller*, *actuator* yang sesuai untuk pengendalian tekanan pada aliran uap.
- b. Bagaimana performansi sistem pengendalian tekanan pada labu pemanas?

1.3 Batasan Masalah

Untuk memfokuskan penyelesaian masalah pada penelitian tugas akhir ini maka batasan masalah yang diangkat adalah sebagai berikut :

- a. Alat yang dirancang dan dibangun hanya memiliki fungsi untuk mengukur nilai temperatur dan tekanan yang dihasilkan oleh *steam*.
- b. Komponen dari alat ini adalah sensor MPX505GP sebagai sensor tekanan, ATmega16 sebagai mikrokontroller dan motor servo sebagai aktuator.

1.4 Tujuan

Tujuan utama dari rancang bangun alat ini adalah untuk memenuhi mata kuliah tugas akhir sebagai syarat kelulusan dari departemen teknik instrumentasi, serta untuk memeberikan solusi pada rumusan masalah yaitu:

- a. Merancang sistem pengendalian tekanan pada aliran uap.
- b. Mendapatkan performansi sistem pengendalian tekanan pada labu pemanas.

1.5 Manfaat

Manfaat dari tugas akhir ini adalah sebagai sistem pengendalian tekanan pada labu pemanas dengan menggunakan

sensor *MPX5050GP* dan ATmega 16 sebagai mikrokontrol serta mengetahui performansi pengendalian yang baik pada alat tersebut.

1.6 Sistematika Laporan

Sistematika laporan yang digunakan dalam penyusunan laporan tugas akhir ini adalah sebagai berikut :

BAB I PENDAHULUAN

Pada bab ini membahas mengenai latar belakang, rumusan masalah, tujuan, batasan masalah, manfaat dan sistematika penulisan dalam tugas akhir ini.

BAB II TINJAUAN PUSTAKA

Pada bab ini membahas mengenai teori-teori penunjang yang diperlukan dalam merealisasikan tugas akhir yaitu berupa teori tentang pengendalian tekanan, steam yang dihasilkan dari proses penguapan, dan perangkat-perangkat yang digunakan dalam pembuatan tugas akhir ini.

BAB III PERANCANGAN DAN PEMBUATAN ALAT

Pada bab ini diuraikan tentang penjelasan mengenai perancangan dan pembuatan alat.

BAB IV PENGUJIAN DAN ANALISIS DATA

Pada bab ini memuat tentang hasil pengujian dari perangkat yang dibuat beserta pembahasannya.

BAB V PENUTUP

Pada bab ini memuat tentang kesimpulan dan saran dari pembuatan tugas akhir ini.

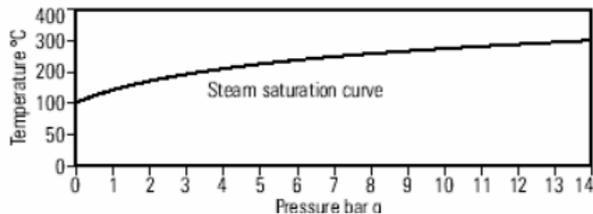
(Halaman ini sengaja dikosongkan)

BAB II TINJAUAN PUSTAKA

2.1 *Steam*

Steam atau uap merupakan gas yang dihasilkan dari proses yang disebut penguapan. Bahan baku yang digunakan untuk menghasilkan *steam* adalah air bersih. Dengan meningkatnya suhu dan air telah mendekati kondisi didihnya, beberapa molekul mendapatkan energi kinetik yang cukup untuk mencapai kecepatan yang membuatnya sewaktu-waktu lepas dari cairan ke ruang di atas permukaan, sebelum jatuh kembali ke cairan. Pemanasan lebih lanjut menyebabkan eksitasi lebih besar dan sejumlah molekul dengan energi cukup untuk meninggalkan cairan jadi meningkat. Dengan mempertimbangkan struktur molekul cairan dan uap, dapat diambil kesimpulan bahwa densitas *steam* lebih kecil dari air, karena molekul *steam* terpisah jauh satu dengan yang lain.

Jika jumlah molekul yang meninggalkan permukaan cairan lebih besar dari yang masuk kembali, maka air akan menguap dengan bebas. Pada keadaan ini air telah mencapai titik didihnya atau suhu jenuhnya, yang dijenuhkan oleh energi panas. Jika tekanannya tetap, penambahan lebih banyak panas tidak mengakibatkan kenaikan suhu lebih lanjut namun menyebabkan air membentuk *steam* jenuh, tetapi jika tekanannya bertambah, maka akan ada penambahan lebih banyak panas dan peningkatan suhu tanpa perubahan fase. Kenaikan tekanan akan meningkatkan entalpi air dan suhu jenuhnya. Hubungan antara suhu jenuh dan tekanan dikenal sebagai kurva *steam* jenuh.



Gambar 2.1 Kurva *Steam* Jenuh [1]

Steam atau air yang berbentuk gas merupakan media panas yang sangat penting karena memiliki kandungan panas yang sangat besar, *steam* merupakan bahan pemanas yang paling banyak digunakan dalam industri. Sebagian besar kandungan panas *steam* merupakan panas kondensasi, karena itu panas tersebut harus dimanfaatkan.

Air sangat menguntungkan jika digunakan sebagai media pemanas karena memiliki panas kondensasi yang besar, tidak mudah terbakar, dan tidak beracun. *Steam* dibuat dipusat pembangkitan *steam* didalam ketel uap dengan menggunakan bahan bakar batu bara, minyak pemanas, atau listrik. Sehingga terbentuk *steam* pada *temperature* yang sesuai dengan tekanan didalam ketel uap. Alat pemanas yang menggunakan *steam* sebagai media pemanas mudah untuk diatur dengan baik. *Steam* dapat dibedakan menjadi tiga macam, yaitu:

- a. *Saturated steam*, yaitu uap air yang terbentuk pada suhu didih dan tidak mengandung titik-titik air maupun gas asing.
- b. *Wet steam*, yaitu campuran dari *saturated steam* dan titik-titik air yang terdistribusi merata. *Steam* ini terbentuk misalnya pada waktu air mendidih dengan sangat kuat atau karena kondensasi sebagian dari uap jenuh.
- c. *Superheated steam*, yaitu uap yang dipanaskan melebihi temperatur didihnya. Pada tekanan yang sama *steam* ini memiliki kerapatan lebih rendah daripada *saturated steam*.

Tabel uap terbagi atas dua bagian, yaitu tabel uap jenuh (*saturated steam*) dan tabel uap lanjut (*superheated steam*). Masing-masing tabel uap tersebut memuat besaran-besaran berikut: tekanan (P), temperatur (T), massa jenis (ρ), volume spesifik (v), entalpi spesifik (h), entropi spesifik (s). [1]

2.2 Labu Didih

Labu didih adalah alat laboratorium yang terbuat dari gelas (*Glass Ware*) dengan bentuk seperti labu. Fungsi labu didih adalah sebagai tempat untuk memanaskan, menyimpan dan mencampur larutan. Labu didih ini memiliki berbagai macam jenis leher, mulai dari *single track*, *double track* hingga *triple track*. Macam bentuknya juga beragam mulai dari yang bundar (*round bottom*) digunakan untuk memanaskan bahan dan ada juga rata (*flat bottom*) digunakan untuk memanaskan bahan dan juga digunakan untuk menyimpan bahan karena saat diletakan di meja posisinya akan lebih stabil. Labu didih biasanya terbuat dari kaca tahan panas pada suhu 120-300 °C. Ukurannya beragam, mulai dari 250 mL sampai 2000 mL. [2]



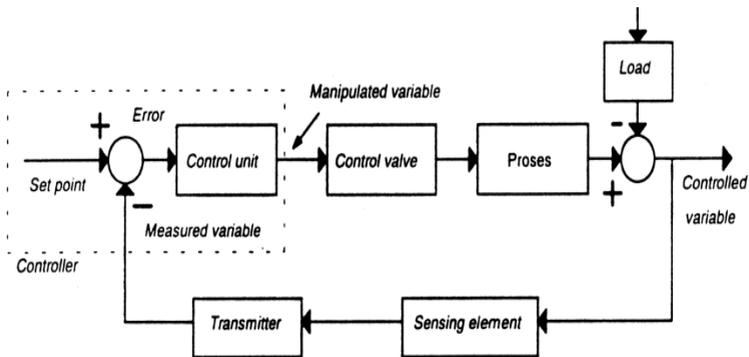
Gambar 2.2 Labu Didih [2]

2.3 Sistem Pengendalian Otomatis

Instrumentasi adalah ilmu yang mempelajari tentang penggunaan peralatan atau instrument untuk mengukur dan mengatur suatu besaran baik kondisi fisis maupun kimia. Dari definisi tersebut dapat diambil suatu kesimpulan atau prinsip dasar, bahwa instrumentasi terdiri dari dua pokok kegiatan yaitu

mengukur dan mengatur suatu besaran. Operasi di industri seperti kilang minyak (*refinery*) dan petrokimia (*petrochemical*) sangat bergantung pada pengukuran dan pengendalian besaran proses. Beberapa besaran proses yang harus diukur dan dikendalikan pada suatu industri proses, misalnya aliran (*flow*) di dalam pipa, tekanan (*pressure*) di dalam sebuah *vessel*, suhu (*temperature*) di *heat exchanger*, serta permukaan (*level*) zat cair di sebuah tangki.

Sistem pengendalian bertujuan untuk mengontrol proses atau *plant* yang terjadi di industri. Operasi yang berjalan di industri ini dapat bersifat diskret (*on-off*), misal *valve* terbuka atau tertutup, motor hidup atau mati, konveyor jalan atau berhenti, dan lain-lain. Selain itu juga secara kontinyu, suatu pengaturan untuk mempertahankan tinggi cairan dalam tangki pada nilai tertentu.



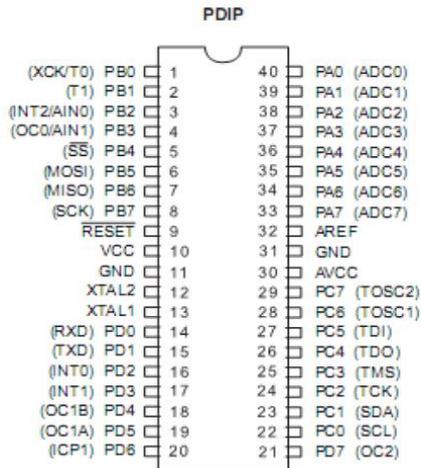
Gambar 2.3 Diagram Blok Pengendalian Otomatis [3]

Di dalam diagram blok sistem pengendalian otomatis, terdapat komponen-komponen pokok seperti elemen proses, elemen pengukuran (*sensing element* dan *transmitter*), elemen *controller* (*control unit*) dan *final control element*. Di dalam gambar bagian atas *controller* mempunyai *summing junction* dengan tanda positif dan negatif. Di titik inilah langkah membandingkan dilakukan dengan mengurangi besaran *set point* dengan sinyal *measurement variable*. Hasilnya adalah sinyal *error*. [3]

2.4 Mikrokontroler ATmega 16

Mikrokontroler adalah sebuah sistem komputer lengkap dalam satu serpih (*chip*). Mikrokontroler lebih dari sekedar sebuah mikroprosesor karena sudah terdapat atau berisikan ROM (*Read-Only Memory*), RAM (*Read-Write Memory*), beberapa bandar masukan maupun keluaran, dan beberapa *peripheral* seperti pencacah/pewaktu, ADC (*Analog to Digital converter*), DAC (*Digital to Analog converter*) dan serial komunikasi. Salah satu mikrokontroler yang banyak digunakan saat ini yaitu mikrokontroler AVR. AVR adalah mikrokontroler RISC (*Reduce Instruction Set Compute*) 8 bit berdasarkan arsitektur Harvard. Secara umum mikrokontroler AVR dapat dikelompokkan menjadi 3 kelompok, yaitu keluarga AT90Sxx, ATmega dan ATtiny. Pada dasarnya yang membedakan masing-masing kelas adalah memori, *peripheral*, dan fiturnya.

Konfigurasi pena (*pin*) mikrokontroler Atmega16 dengan kemasan 40-pena dapat dilihat pada Gambar 2.10. Dari gambar tersebut dapat terlihat ATmega16 memiliki 8 pena untuk masing-masing bandar A (*Port A*), bandar B (*Port B*), bandar C (*Port C*), dan bandar D (*Port D*). [4]



Gambar 2.4 Pin-pin ATmega 16 [4]

2.5 Motor Servo

Motor servo adalah sebuah perangkat atau aktuator putar (motor) yang dirancang dengan sistem kontrol umpan balik loop tertutup (servo), sehingga dapat di set-up atau di atur untuk menentukan dan memastikan posisi sudut dari poros output motor untuk menggerakkan *valve*. Motor servo merupakan perangkat yang terdiri dari motor DC, serangkaian gear, rangkaian kontrol dan potensiometer. Serangkaian *gear* yang melekat pada poros motor DC akan memperlambat putaran poros dan meningkatkan torsi motor servo, sedangkan potensiometer dengan perubahan resistansinya saat motor berputar berfungsi sebagai penentu batas posisi putaran poros motor servo.



Gambar 2.5 Motor Servo MG 996R Tower Pro [5]

Ada dua jenis motor servo, yaitu motor servo AC dan DC. Motor servo AC lebih dapat menangani arus yang tinggi atau beban berat, sehingga sering diaplikasikan pada mesin-mesin industri. Sedangkan motor servo DC digunakan pada aplikasi-aplikasi yang lebih kecil. Dan bila dibedakan menurut rotasinya, umumnya terdapat dua jenis motor servo yang terdapat di pasaran, yaitu *motor servo rotation 180°* dan *servo rotation continuous 360°*. [5]

- a. Motor servo *standard (servo rotation 180°)* adalah jenis yang paling umum dari motor servo, dimana putaran poros outputnya terbatas hanya 90° kearah kanan dan 90° kearah kiri. Dengan kata lain total putarannya hanya setengah lingkaran atau 180° .

- b. Motor servo *rotation continuous* 360° merupakan jenis motor servo yang sama dengan jenis servo *standard*, perputaran porosnya tanpa batasan dengan kata lain dapat berputar terus, baik ke arah kanan maupun kiri.

2.6 Sensor Tekanan

Secara fisis tekanan dirumuskan sebagai gaya yang diberikan suatu cairan atau gas terhadap suatu permukaan. Sensor yang bersentuhan langsung dengan materi yang di deteksi menghasilkan sinyal, dalam hal ini sensor tekanan bertindak sebagai transduser. Sinyal yang dihasilkan sebanding dengan besarnya tekanan materi yang dideteksi terhadap sensor tersebut. Ada tiga jenis tipe dari pengukuran tekanan yaitu tekanan absolut, tekanan diferensial, dan tekanan gauge. Pada tekanan absolut mengukur di daerah vakum. Pada pengukuran menggunakan tekanan diferensial adalah perbedaan tekanan diantara dua jenis tekanan yang akan diukur. Sedangkan pada tekanan gauge adalah dengan menggunakan tekanan referensi contohnya adalah tekanan darah (Singh, 2002).



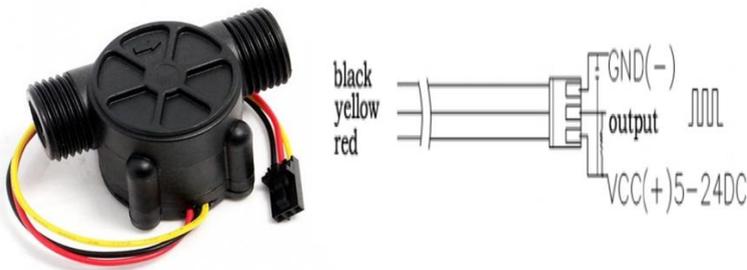
Gambar 2.6 Sensor Tekanan MPX5050GP[6]

Sensor tekanan tipe MPX5050GP ini mampu mendeteksi tekanan sebesar 0 sampai dengan 50 kPa. MPX5050GP hanya membutuhkan *supply* tegangan +5 Volt. Seperti sensor tekanan pada umumnya, sensor akan mengubah tekanan menjadi tegangan. Semakin besar tekanan yang diberikan, semakin besar pula tegangan yang dihasilkan. Sensor ini dilengkapi *chip signal*

conditioned dimana keluaran dari sensor ini tidak perlu dikuatkan lagi. [6]

2.7 Water Flow Sensor

Water flow sensor adalah alat untuk mengukur jumlah atau laju aliran dari suatu fluida yang mengalir dalam pipa atau sambungan terbuka. Alat ini terdiri dari *primary device*, yang disebut sebagai alat utama dan *secondary device* (alat bantu sekunder). Alat utama menghasilkan suatu signal yang merespon terhadap aliran karena laju aliran tersebut telah terganggu. *Water flow* sensor terdiri dari tubuh katup plastik, rotor air dan hall efek. Prinsip kerja *water flow* sensor ini adalah dengan memanfaatkan fenomena efek hall yaitu efek medan magnetik terhadap partikel bermuatan yang bergerak. Ketika ada arus listrik yang mengalir pada divais efek hall yang ditempatkan dalam medan magnet yang arahnya tegak lurus arus listrik, pergerakan pembawa muatan akan berbelok ke salah satu sisi dan menghasilkan gaya Lorentz yang bekerja pada partikel menjadi nol. Perbedaan potensial antara kedua sisi divais tersebut disebut potensial hall. Potensial hall ini sebanding dengan medan magnet dan arus listrik yang melalui device.



Gambar 2.7 Skematik Water Flow Sensor G ½ [7]

Water Flow sensor terdiri dari tubuh katup plastik, rotor air, dan sensor hall efek. Ketika air mengalir melalui gulungan rotor-rotor. Kecepatan perubahan dengan tingkat yang berbeda aliran. Sesuai sensor hall efek output sinyal pulsa. Kelebihan sensor ini

adalah hanya membutuhkan 1 sinyal (SIG) selain jalur 5V dc dan Ground. [7]

2.8 LCD

Liquid Crystal Display (LCD) adalah suatu alat untuk *display* berbagai *character*. LCD ini mempunyai beberapa ukuran mengikuti bilangan *character* seperti 4x20. 4x20 bermakna LCD tersebut mempunyai 4 baris dan 20 kolom. *LCD (Liquid Cristal Display)* berfungsi untuk menampilkan karakter angka, huruf atau simbol dengan baik dan konsumsi arus yang lebih rendah. *LCD (Liquid Cristal Display)* dot matrik terdiri dari bagian penampil karakter (*LCD*) yang berfungsi untuk menampilkan karakter dan bagian sistem prosesor *LCD* dalam bentuk rangkaian modul dengan mikrokontroler yang diletakan dibagian belakang *LCD* tersebut yang berfungsi untuk mengatur tampilan LCD serta mengatur komunikasi antara LCD dengan mikrokontroler. Modul prosesor pada LCD memiliki memori tersendiri sebagai berikut: *CGROM (Character Generator Read Only Memory)*, *CGRAM (Character Generator Random Access Memory)*, *DDRAM (Display Data Random Access Memory)*.

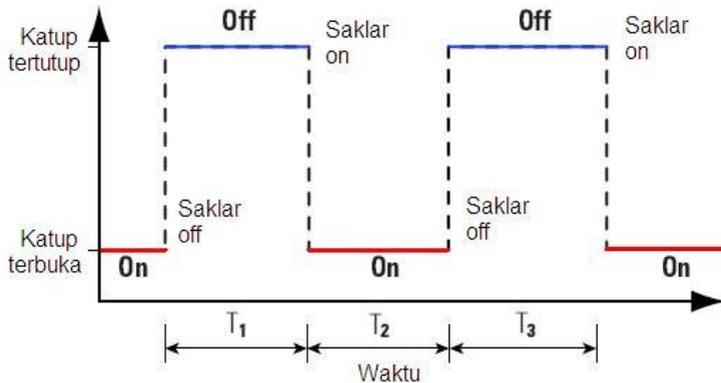


Gambar 2.8 LCD *Character* 4x20

LCD karakter dalam pengendaliannya cenderung lebih mudah dibandingkan dengan LCD grafik. Namun ada kesamaan diantara keduanya, yaitu inisialisasi. Inisialisasi adalah prosedur awal yang perlu dilakukan dan dikondisikan kepada LCD agar LCD dapat bekerja dengan baik. [8]

2.9 Pengendalian *ON-OFF*

Karakteristik kontroler on – off ini hanya bekerja pada 2 posisi, yaitu on dan off. Kerja kontroler on – off banyak digunakan pada aksi pengontrolan yang sederhana karena harganya murah. Karena sistem kerja yang digunakan adalah on – off saja, hasil output dari sistem pengendalian ini akan menyebabkan proses variabel tidak akan pernah konstan. Besar kecilnya fluktuasi process variabel ditentukan oleh titik dimana kontroler dalam keadaan on dan off. Pengendalian dengan aksi kontrol ini juga menggunakan feedback [3].



Gambar 2.9 Aksi Kontrol On Off [3]

2.10 Karakteristik Statik

Karakteristik statik pengukuran merupakan karakteristik yang ditentukan melalui perhitungan matematik atau secara grafik. Karakteristik statik merupakan karakter yang menggambarkan parameter dari sebuah instrument pada saat keadaan *steady*. [9] Karakteristik statik terdiri dari:

a. *Range*

Range merupakan selisih antara nilai minimum dan maksimum yang terukur oleh suatu instrument atau alat ukur.

b. Akurasi

Akurasi merupakan tingkat ketelitian suatu alat dalam memberikan hasil pengukuran. **Toleransi**

Toleransi menunjukkan kesalahan maksimum yang diperbolehkan pada hasil pembacaan alat ukur.

c. Sensitivitas

Sensitivitas menunjukkan perubahan output instrument yang terjadi saat diberi kualitas pengukuran yang berbeda. Sensitivitas diperoleh dari $\Delta O / \Delta I$, dimana O adalah *output* dan I adalah *input*.

d. Repeatability

Repeatability merupakan kemampuan *instrument* dalam menampilkan ulang *output* pengukuran yang sama pada pengukuran yang berulang.

e. Linearitas

Linearitas merupakan grafik yang menampilkan pengukuran nilai sebenarnya (*input*) yang dapat menghasilkan *output* nilai yang ditunjukkan oleh instrument (*output*). Pengukuran yang baik adalah ketika input pengukuran dan output pengukuran berbanding lurus (*linear*). Linearitas dapat diperoleh dari persamaan berikut:

$$O - O_{min} = \frac{O_{max} - O_{min}}{I_{max} - I_{min}} (I - I_{min}) \dots\dots\dots 2.1$$

Persamaan Linearitas :

$$O_{ideal} = KI + a \dots\dots\dots 2.2$$

Dimana:

$$K = \frac{O_{max} - O_{min}}{I_{max} - I_{min}} \dots\dots\dots 2.3$$

Dan

$$a = O_{min} - KI_{min} \dots\dots\dots 2.4$$

Dimana :

$$O = 1.6 \times 10^{-3}I + 4.0 \dots\dots\dots 2.5$$

2.11 Teori Ketidakpastian

Ketidakpastian pengukuran merupakan tingkat seberapa besar ketidakpastian yang dihasilkan oleh suatu alat ukur. Dalam menghitung ketidakpastian pengukuran ada beberapa langkah yang harus dihitung, antara lain:

a. Koreksi

Koreksi dapat diperoleh dengan persamaan berikut

$$\text{Koreksi} = \text{Pembacaan standard-Pembacaan alat} \dots\dots 2.6$$

b. Standard deviasi

$$\sigma = \frac{\sqrt{\sum(Di - D')^2}}{n-1} \dots\dots\dots 2.7$$

dimana :

Di = koreksi alat ukur

Di' = rata-rata koreksi

n = Banyak range pengukuran

c. Analisa Type A, (Ua)

Pada analisa tipe A ini hasilnya diperoleh dari data pengukuran. Adapun persamaannya adalah sebagai berikut:

$$Ua1 = \frac{\sigma_{maks}}{\sqrt{n}} \dots\dots\dots 2.8$$

$$Ua2 = \sqrt{\frac{SSR}{n-2}} \dots\dots\dots 2.9$$

d. Analisa Type B, (Ub)

Analisa tipe B ini diperoleh berdasarkan sertifikat kalibrasi atau spesifikasi dari alat ukur. Adapun persamaannya adalah sebagai berikut:

$$Ub1 = \frac{\text{Resolusi} / 2}{\sqrt{3}} \dots\dots\dots 2.10$$

Dimana :

SSR = Sum Square Residual

$$Ub2 = \frac{a}{k} \dots\dots\dots 2.11$$

e. Ketidakpastian Kombinasi (UC)

Uc merupakan Ketidakpastian kombinasi dari ketidakpastian tipe A dan ketidakpastian tipe B. Adapun persamaan dari ketidakpastian kombinasi adalah:

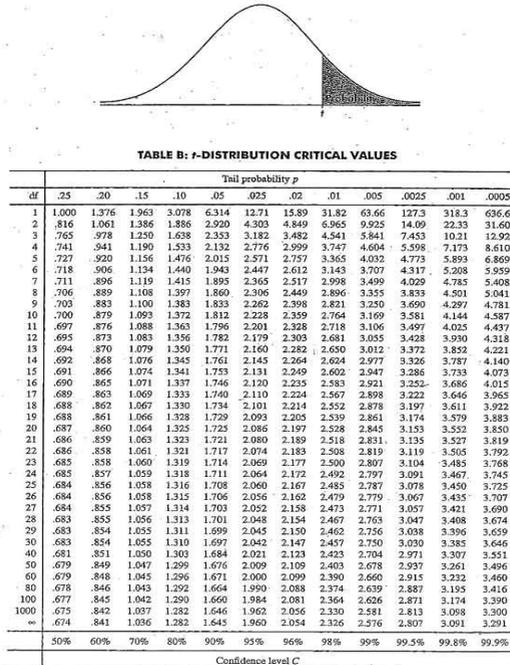
$$UC = \sqrt{Ua1^2 + Ua2^2 + Ub1^2 + Ub2^2} \dots\dots\dots 2.12$$

f. Ketidakpastian Diperluas

Hasil akhir kalibrasi adalah ketidakpastian diperluas sehingga alat ukur tersebut dapat diketahui ketidakpastiannya melalui *Uexpand*. Persamaan *Uexpand* adalah:

$$U_{expand} = k.Uc \dots\dots\dots 2.13$$

Untuk mencari nilai k, maka melihat table t student sesuai dengan *confidence level* 95%. Tabel *T student* dapat dilihat pada gambar 2.15



Gambar 2.10 Tabel *T-student*[9]

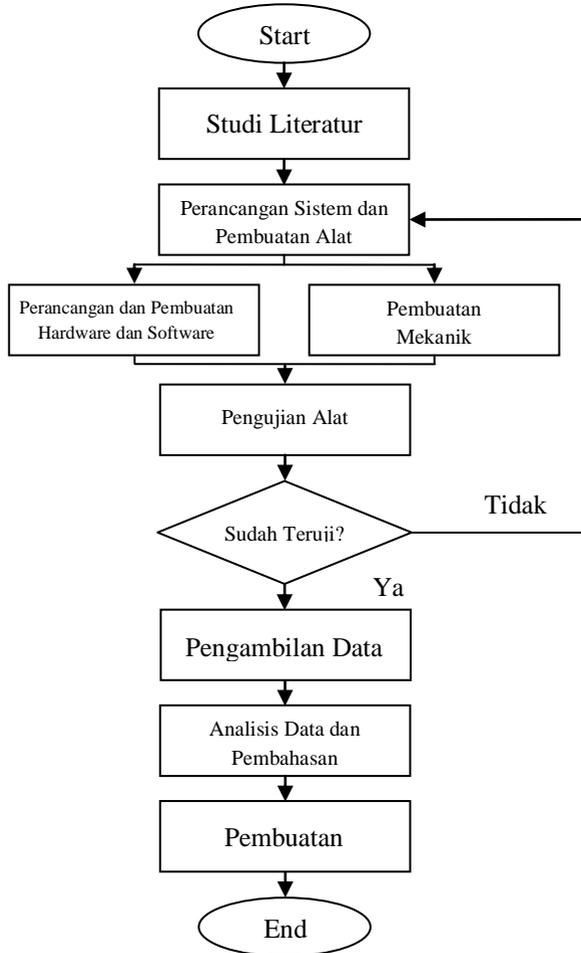
g. V effektiv

$$V_{eff} = \frac{(U_c)^4}{\frac{\sum (u_i)^4}{V_i}} \dots\dots\dots 2.14$$

BAB III PERANCANGAN DAN PEMBUATAN ALAT

3.1 Diagram Alir dan Diagram Blok Perancangan Alat

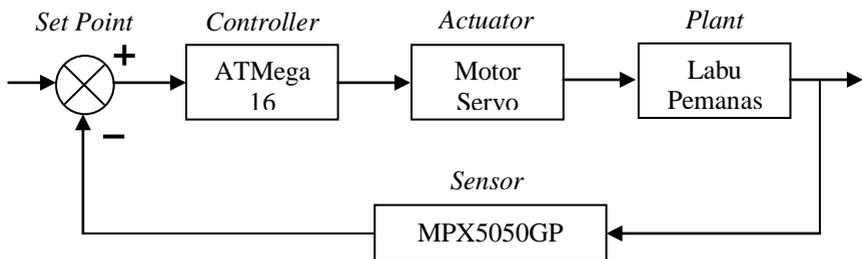
Tahapan penelitian Tugas Akhir ini, secara umum dapat digambarkan dalam *flowchart* seperti dibawah ini:



Gambar 3.1 Diagram Alir Tugas Akhir

Diagram alir diatas merupakan tahapan pengerjaan tugas akhir mulai dari start hingga end. Tahap awal pada diagram alir ini dimulai dengan adanya studi literatur sebagai upaya pemahaman terhadap materi yang menunjang tugas akhir. Setelah melakukan studi literatur, selanjutnya adalah melakukan perancangan sistem dan mempersiapkan komponen yang dibutuhkan. Kemudian dibuat perancangan *hardware*, *software*, dan mekanik dari sistem pengendalian temperatur berbasis ATMega16. Setelah itu dibuat sistem pengendalian temperatur dan tekanan, kemudian dilakukan pengujian alat dengan memberi *input* berupa temperatur dan tekanan pada labu pemanas, sehingga dapat diketahui pembacaan temperatur dan tekanan. Apabila semua rancang bangun sistem pengendalian temperatur dan tekanan pada labu pemanas dapat bekerja dengan baik. Setelah pembuatan rancangan telah selesai dengan hasil yang sesuai dengan yang diinginkan, kemudian dilakukan analisis data dengan memanfaatkan hasil dari uji performansi dan sistem pengendalian. Setelah semua hasil yang diinginkan tercapai mulai dari studi literatur hingga analisa data dan kesimpulan dicantumkan dalam sebuah laporan.

Diagram blok merupakan salah satu cara yang paling sederhana untuk menjelaskan cara kerja dari suatu sistem. Dengan diagram blok dapat menganalisa cara kerja rangkaian dan merancang hardware yang akan dibuat secara umum. Adapun diagram blok dari sistem yang dirancang, seperti yang diperlihatkan pada gambar 3.2

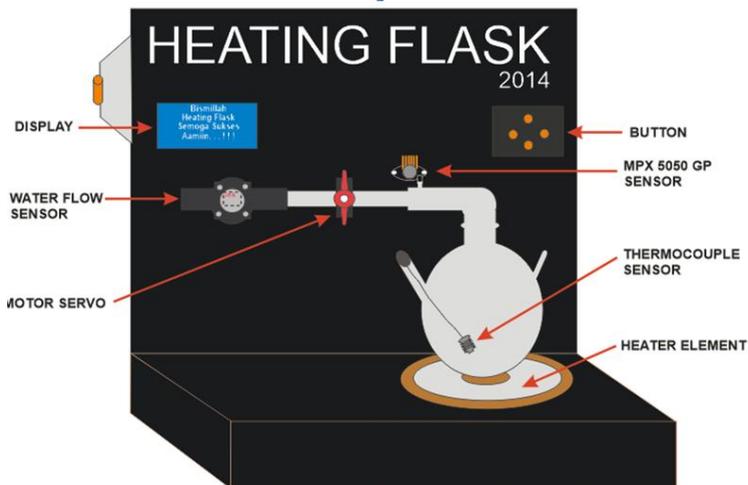


Gambar 3.2 Diagram Blok Sistem Pengendalian Tekanan

Gambar 3.2 diatas merupakan gambar diagram blok sistem pengendalian tekanan pada labu pemanas yang terdiri dari mikrokontroler ATmega 16 sebagai *controller*, motor servo sebagai *actuator*, labu pemanas sebagai tempat pemrosesan, dan sensor MPX5050GP sebagai sensor untuk sensing tekanan didalam labu pemanas (dalam satuan KPa). Nilai tekanan yang dibaca oleh sensor akan ditampilkan ke *display* LCD (*Liquid Crystal Display*) 4x20 dalam satuan KPa.

3.2 Gambaran Umum

Sistem pengendalian tekanan pada *plant* labu pemanas ini terletak pada aliran uap. Keadaan awal *valve* adalah tertutup atau *normally close* karena tekanan pada *steam* yang dihasilkan oleh proses penguapan pada labu pemanas belum merupakan tekanan yang sesuai dengan *set point*, sehingga diperlukan adanya sensor tekanan yaitu MPX5050GP. Apabila tekanan sudah mencapai *set point* maka motor servo akan menggerakkan *valve* untuk membuka *valve*. Sedangkan apabila tekanan kurang dari *set point* maka motor servo akan menggerakkan *valve* untuk menutup *valve*.



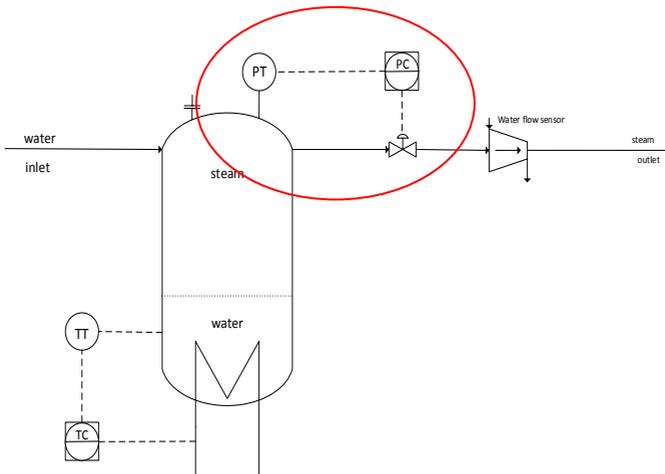
Gambar 3.3 Desain *Plant* Labu Pemanas

3.3 Studi Literatur Alat Ukur Tekanan dan Metode Kontrol Tekanan

Tahap awal pada pengerjaan tugas akhir ini dimulai dengan adanya studi literatur sebagai upaya pemahaman terhadap materi yang menunjang tugas akhir mengenai "Rancang Bangun Sistem Pengendalian Tekanan pada Aliran Uap". Studi literatur ini dilakukan dengan mencari dan mempelajari informasi dari *e-book* maupun manual *book* mengenai pengendalian tekanan. Selain belajar sistem pada perancangan sistem control, juga dilakukan mencari literature-literatur yang berkaitan dengan elemen-elemen yang digunakan dalam pembuatan sistem pengendalian tekanan misalnya *datasheet* pada sensor, *actuator*, dan mikrokontroler.

3.4 Perancangan Sistem dan Pembuatan Hardware serta Software Alat Pengendalian Tekanan

Pada perancangan mini plant pengendalian tekanan ini terdapat pada labu didih berukuran 1000 ml dengan material *glass*. *Heater element* diletakan dibawah labu didih untuk memanaskan air sehingga menghasilkan *steam*.



Gambar 3.4 P&ID Sistem Pengendalian Tekanan pada Labu Pemanas

Pada perancangan dan pembuatan *hardware* serta *software* terdapat beberapa tahap yaitu melakukan pengkopelan motor servo dengan *valve*, *wiring* sensor MPX5050GP ke mikrokontroler ATmega 16 untuk pengolahan data dari sensor dan aksi yang akan dilakukan oleh aktuator (*valve*).

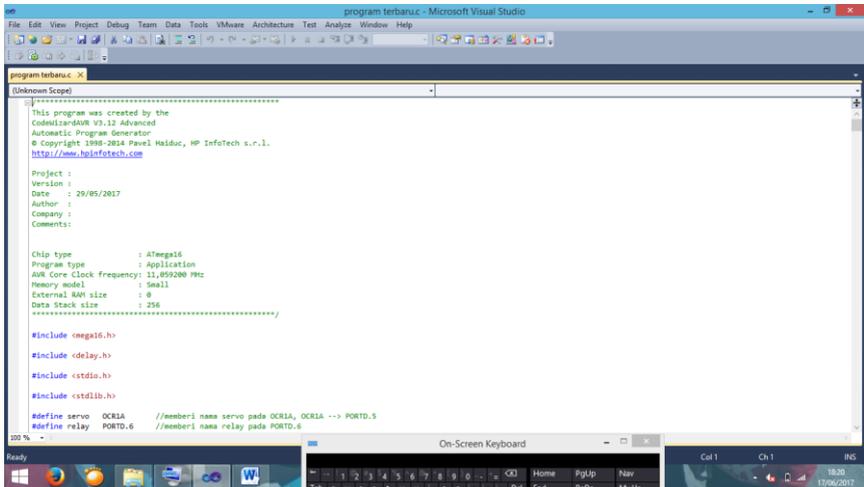
3.4.1 Perancangan dan Pembuatan Alat (*Hardware*)

Alat pengendali tekanan ini menggunakan sensor MPX5050GP. Sensor ini berfungsi untuk menyensing berapa tekanan pada aliran uap yang ada didalam labu pemanas.

Sensor tekanan tipe MPX5050GP ini mampu mendeteksi tekanan sebesar 0 sampai dengan 50 kPa. MPX5050GP hanya membutuhkan *supply* tegangan +5 Volt. Seperti sensor tekanan pada umumnya, sensor akan mengubah tekanan menjadi tegangan. Semakin besar tekanan yang diberikan, semakin besar pula tegangan yang dihasilkan. Sensor ini dilengkapi *chip signal conditioned* dimana keluaran dari sensor ini tidak perlu dikuatkan lagi.

3.4.2 Perancangan dan Pembuatan Alat (*Software*)

Pada perancangan *software* ini merupakan gabungan dari rangkaian pemrosesan sinyal. Rangkaian pemrosesan sinyal pada alat ini menggunakan ATmega16. Untuk membuat sebuah program di ATmega16 dibutuhkan *software* Code Vision AVR. Pada program ini akan dikoding untuk menerima sinyal masukan. Sinyal masukan ini berasal dari sensing *element* yaitu sensor MPX5050GP. Sehingga terbaca berapa tekanan yang sedang diukur pada LCD dengan terlebih dahulu didownload ke ATmega16. Untuk menginstruksi ATmega16 sebagai kontroler, maka dibuat *code* atau *listing program* sesuai dengan instruksi yang ingin diberikan. Berikut merupakan cuplikan dari *code* untuk instruksi ATmega16 pada tugas akhir ini.



Gambar 3.5 Software Code Vision AVR

3.5 Integrasi *Hardware* dan *Controller*

Ketika *coding* program sudah jadi maka dilakukan pengintegrasian antara *hardware* dengan *software*. Didalam *software* dilakukan penyamaan *Port* yang digunakan untuk melakukan proses *uploading* program yang ada ke mikro *chip* ATmega16.

Penyesuaian *interface* antara *software* CodeVision AVR dan *hardware* ATmega16 dapat dikonfigurasi melalui fitur COM. Jika sudah berhasil, maka hasil program dapat dilihat pada LCD.

3.6 Pengujian Sistem Pengukuran dan Sistem Pengendalian

Pengujian ini bertujuan untuk mengetahui apakah program sudah ter-upload ke hardware ATmega16 dan berjalan dengan baik. Pengujian ini dilakukan dengan cara mencoba sensor untuk memulai pengukuran. Apabila sensor masih belum menampilkan data pada LCD, maka proses pemrograman pada CodeVision AVR dan integrasi ke hardware perlu diulang. Sedangkan untuk sistem pengendalian dilakukan dengan cara melihat apakah proses variabel dapat tercapai dan terjaga pada set point. Apabila sistem

pengendalian tidak membuat nilai proses variabel sama dengan set point, maka proses pemrograman dan integrasi software harus diulang.

3.7 Pengambilan dan Analisis Data

Pada tahap pengambilan data untuk karakteristik statik dan kalibrasi merupakan tahap dimana melihat spesifikasi yang dimiliki oleh sistem pengendalian yang telah dibuat, dengan adanya data tersebut dapat diketahui performansi sistem pengendalian. Pada karakteristik statik alat yang dicari yaitu nilai *range*, *span*, resolusi, sensitivitas, *non-linieritas*, *hysteresis*, serta akurasi. Sedangkan untuk data kalibrasi digunakan untuk mencari nilai ketidakpastian dari hasil pengukuran ketika menggunakan perangkat sistem pengendalian untuk arus dan tegangan *input-output* ini. Berikut merupakan langkah-langkah pengambilan data karakteristik statik:

- Nilai *range*, *span*, dan resolusi sistem pengendalian dicatat sesuai spesifikasi sensor arus dan tegangannya.
- Nilai sensitivitas ditentukan dengan rumus:

$$\text{Sensitivitas} = \frac{\Delta O}{\Delta I}$$

- Nilai *linieritas* ditentukan dengan rumus:

$$O - O_{min} = \left[\frac{O_{maks} - O_{min}}{I_{maks} - I_{min}} \right] (I - I_{min})$$

$$O_{ideal} = KI + \alpha$$

Dimana:

$$K = \text{Kemiringan garis lurus ideal} = \frac{O_{maks} - O_{min}}{I_{maks} - I_{min}}$$

$$\alpha = \text{Koefisien garis lurus ideal} = O_{min} - KI_{min}$$

- Histerisis ditentukan dengan melakukan pengambilan data input naik dan turun, dengan persamaan histerisis:

$$H(I) = O(I)_{I\downarrow} - O(I)_{I\uparrow}$$

$$\% \text{ Maksimum Histerisis} = \frac{H}{O_{maks} - O_{min}} \times 100\%$$

- Nilai akurasi ditentukan dari pembacaan alat dengan pembacaan standar, nilai akurasi ditentukan dari nilai kesalahan akurasi dengan persamaannya, yaitu:

$$A = 1 - \left| \text{Rata} - \text{Rata} \frac{(\text{Pemb. std} - \text{Pemb. alat})}{\text{Pemb. std}} \right|$$

3.8 Prosedur Operasional

Untuk mengaktifkan mini plant pemurnian garam ini perlu diperhatikan tata cara operasionalnya, yaitu sebagai berikut :

1. Pastikan semua wiring rangkaian sudah terpasang dengan benar dan baik.
2. Pastikan tidak ada kebocoran pada masing-masing bagian plant termasuk tabung dan perpipaannya.
3. Pastikan pemasangan sensor sudah dilakukan dengan baik dan benar
4. Pastikan sambungan kabel yang terhubung dengan tegangan AC terhubung dengan benar, sesuai dan pastikan tidak ada kabel yang terkelupas.
5. Pastikan apakah power supply untuk kontroller dan bagian-bagian lainnya telah terpasang dan terhubung dengan benar.
6. Hubungkan kabel power ke listrik AC PLN.
7. Lihat apakah ada sistem yang terjadi error. Jika terjadi error maka putuskan kabel dari listrik AC PLN dan lakukan *troubleshooting*

3.9 Penulisan Laporan

Setelah semua hasil yang diinginkan tercapai, kemudian semua hasil mulai dari studi literatur sampai dengan analisa data dan kesimpulan dicantumkan dalam sebuah laporan.

(Halaman ini sengaja dikosongkan)

BAB IV ANALISA DATA DAN PEMBAHASAN

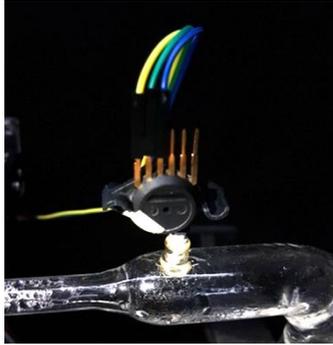
4.1 Pengujian Sensor Tekanan MPX5050GP

Berikut merupakan hasil perancangan sistem pengendalian tekanan pada aliran uap.



Gambar 4.1 Mini Plant Labu Pemanas

Pada proses pengendalian tekanan pada aliran uap ini digunakan sensor tekanan MPX5050GP. Sensor tekanan tipe MPX5050GP ini mampu mendeteksi tekanan sebesar 0 sampai dengan 50 kPa. MPX5050GP hanya membutuhkan *supply* tegangan +5 Volt. Seperti sensor tekanan pada umumnya, sensor akan mengubah tekanan menjadi tegangan. Semakin besar tekanan yang diberikan, semakin besar pula tegangan yang dihasilkan. Sensor ini dilengkapi *chip signal conditioned* dimana keluaran dari sensor ini tidak perlu dikuatkan lagi.



Gambar 4.2 Penempatan Sensor MPX5050GP

Sensor MPX5050GP dipasang pada bagian atas di labu pemanas seperti yang terlihat pada gambar 4.1, PT merupakan sensor MPX5050GP. TT merupakan sensor *thermocouple type K* yang terletak pada bagian bawah di labu pemanas. PC dan TC merupakan kontroler berupa mikrokontroler ATmega16.

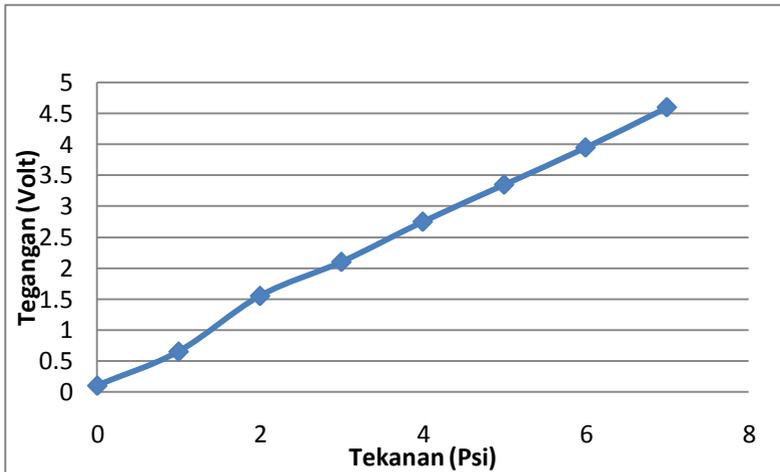
Pengujian sensor dilakukan untuk mengetahui sensitivitas sensor. Pengujian dilakukan dengan mengaktifkan tegangan referensi 5V yang masuk ke sensor kemudian diberi tekanan dari *compressor* yang telah diatur keluarannya menggunakan *regulator pressure*. Adapun tekanan yang terukur adalah tekanan *gauge*.

Tabel 4.1 Hasil Pengujian V_{Out} pada Sensor MPX5050GP

Data ke-	Tekanan Gauge (Psi)	V_{in} (V)	V_{out} P1 (V)	V_{out} P2 (V)	V_{out} Rata-rata (V)
1	0	5	0.1	0.1	0.1
2	1	5	0.7	0.6	0.65
3	2	5	1.6	1.5	1.55
4	3	5	2.1	2.1	2.1
5	4	5	2.7	2.8	2.75
6	5	5	3.3	3.4	3.35
7	6	5	3.9	4	3.95
8	7	5	4.6	4.6	4.6

$$\begin{aligned} \text{Sensitivitas} &= \frac{\Delta \text{Output}}{\Delta \text{Input}} = \frac{(4.6-0.1)V}{(7-0)\text{Psi}} = 0.642857 \text{ V/Psi} \\ &= 642.857\text{mV/Psi} = 93.238488\text{mV/kPa} \end{aligned}$$

Dari data perhitungan sensitivitas yang diperoleh dari tabel 4.1, didapatkan nilai sensitivitas sensor MPX5050GP adalah sebesar 93,238488 mV/kPa.



Gambar 4.3 Grafik Pengujian Sensitivitas Sensor MPX5050GP

Gambar 4.3 merupakan grafik respon keluaran dari sensor MPX5050GP yang diambil dari data pengukuran. Data keluaran berupa tegangan pada sumbu Y terhadap data masukan yang berupa tekanan dengan *range* 0-7 Psi pada sumbu X. berdasarkan gambar 4.3, grafik menunjukkan bahwa keluaran sensor MPX5050GP adalah *linear* terhadap masukan yang berupa *pressure*. Hal ini dikarenakan sensor MPX5050GP merupakan sensor yang mempunyai sensitivitas *linear* yang sudah terkalibrasi.

Tabel 4.2 Data Pengujian Naik Sensor MPX5050GP (KPa)

No.	Pembacaan Standar, Xi (Psi)	Pembacaan Alat Naik (KPa)					Rata-rata (Psi)
1	0	0	0	0	0	0	0
2	1	6.75	6.7	6.56	6.74	6.74	6.698
3	2	14.4	14.4	14.34	14.3	14.44	14.382
4	3	21.6	21.4	21.45	21.3	21.1	21.366
5	4	28	28	28.04	28	28.06	28.024
6	5	35.3	35.2	35.18	35.2	35	35.164
7	6	41.8	41.8	41.66	41.8	41.81	41.762
8	7	49.6	49.5	49.52	49.5	49.44	49.492

Tabel 4.3 Data Pengujian Naik Sensor MPX5050GP (Psi)

No.	Pembacaan Standar, Xi (Psi)	Pembacaan Alat Naik (Psi)					Rata-rata (Psi)
1	0	0	0	0	0	0	0
2	1	0.98	0.971	0.951	0.98	0.977	0.971
3	2	2.09	2.088	2.079	2.07	2.094	2.0854
4	3	3.13	3.1	3.11	3.09	3.06	3.0982
5	4	4.06	4.063	4.066	4.06	4.069	4.064
6	5	5.11	5.105	5.102	5.1	5.076	5.0996
7	6	6.06	6.055	6.042	6.06	6.064	6.0568
8	7	7.19	7.173	7.182	7.17	7.17	7.1778
Jumlah	28						28.5528
Rata-rata	3.5						3.5691

4.2 Data Spesifikasi Alat

Karakteristik statik adalah karakteristik yang harus diperhatikan apabila alat tersebut digunakan untuk mengukur suatu kondisi yang tidak berubah karena waktu atau hanya berubah secara lambat laun. Untuk itu perlu dilakukan perhitungan untuk mengetahui nilai karakteristik dari sensor tekanan MPX5050GP diantaranya sebagai berikut:

Tabel 4.6 Pengambilan Data Naik dan Turun pada Sensor MPX5050GP

No	Data Naik		Data Turun		H (I) (Psi)	(P.std- P.alat)/ P.std	O Ideal	Non- Linieritas Per Input
	Pin (Psi)	Pout (Psi)	Pin (Psi)	Pout (Psi)				
1	0	0	0	0	0	0	0	0
2	1	0.971	1	0.9044	0.009278609	0.029	1.0254	0.0544
3	2	2.0854	2	2.0982	0.001783276	0.043	2.0508	0.0346
4	3	3.0982	3	3.0558	0.005907102	0.033	3.0762	0.022
5	4	4.064	4	4.0622	0.000250773	0.016	4.1016	0.0376
6	5	5.0996	5	5.0808	0.002619187	0.02	5.127	0.0274
7	6	6.0568	6	6.095	0.005321965	0.009	6.1524	0.0956
8	7	7.1778	7	7.0054	0.024018501	0.025	7.1778	0
jumlah		28.553		28.302		0.175		
Rata-		3.5691		3.5377		0.0219		

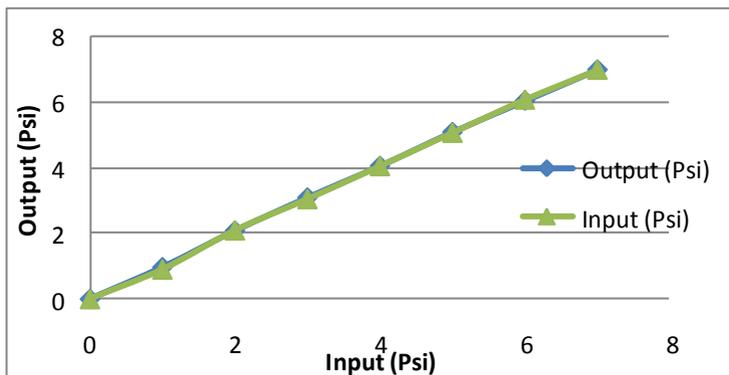
Berikut ini hasil perhitungan nilai karakteristik statik tekanan berdasarkan data pada tabel 4.6:

a. Histerisis

$H(I) = O(I)_{I\downarrow} - O(I)_{I\uparrow}$, $\hat{H} = H(I)_{max}$, sehingga:

$$\begin{aligned} \% \text{ Maksimum Histerisis} &= \frac{\hat{H}}{O_{max} - O_{min}} \times 100\% \\ &= \frac{0.024018501}{7.1778 - 0} \times 100\% \\ &= 0.3346 \% \end{aligned}$$

Berikut gambar grafik histerisis dari data pengukuran naik dan turun tersebut:



Gambar 4.4 Grafik Histerisis Sensor MPX5050GP

Gambar 4.4 menunjukkan grafik histerisis pada sensor MPX5050GP, dimana dari grafik tersebut diketahui bahwa histerisis tertinggi terdapat ketika *input pressure* bernilai 7 Psi. hal tersebut dikarenakan adanya nilai koreksi antara pembacaan naik dan pembacaan turun sebesar 0.0240185.

b. Non-linieritas

$$O_{ideal} = KI + \alpha$$

Dengan K adalah kemiringan garis:

$$\begin{aligned} K &= \frac{O_{max} - O_{min}}{I_{max} - I_{min}} \\ &= \frac{7.1778 - 0}{7 - 0} \\ &= 1.0254 \end{aligned}$$

Dan α adalah pembuat nol (*zero bias*):

$$\begin{aligned} \alpha &= O_{min} - KI_{min} \\ &= 0 - (1.0254 \times 0) \\ &= 0 \end{aligned}$$

Sehingga didapatkan nilai Non-linieritas:

$$\begin{aligned} \dot{N} &= \frac{[O - KI + \alpha]_{max}}{O_{max} - O_{min}} \times 100\% \\ &= \frac{0.0956}{7.1778 - 0} \times 100\% \\ &= 1.3318844 \% \end{aligned}$$

c. Akurasi

$$MAPE = \frac{(100\%)}{n} \sum_{t=1}^n \frac{|X_t - F_t|}{X_t}$$

dengan

X_t = Data aktual pada periode t

F_t = Data pemodelan pada periode t

n = Jumlah data

$$\begin{aligned} MAPE &= \frac{(100\%)}{10} \times 0.175 \\ &= 1.75\% \end{aligned}$$

Nilai akurasi berarti ketakakuratan (*inaccuracy*), yaitu selisih maksimum antara nilai keluaran sensor dari nilai masukan ideal/sesungguhnya (*actual input*). Nilai akurasi ini diperoleh dengan menggunakan persamaan MAPE (*Mean Absolute Percentage Error*) yaitu salah satu metode yang dapat digunakan untuk menghitung validasi.

Sehingga dihasilkan nilai:

Range	: Pressure 0-7 Psi
Span	: Pressure 7 Psi
Resolusi	: 0.01
Sensitivitas (K)	: 93.238488 mV/kPa (Dari data tabel 4.1)
Histerisis	: 0.3346 %
Non-linieritas	: 1.33188 %
Akurasi	: 1.75 %

Berikut ini merupakan hasil pengukuran kalibrasi untuk mencari nilai ketidakpastian alat ukur:

Tabel 4.7 Data Kalibrasi pada Sensor MPX5050GP (A)

No.	Pembacaan Standar, Xi (Psi)	Pembacaan Alat		Rata-rata (Psi)	Koreksi, Yi (Psi)
		Rata-rata Naik (Psi)	Rata-rata Turun (Psi)		
1	0	0	0	0	0
2	1	0.971	0.9044	0.9377	0.0623
3	2	2.0854	2.0982	2.0918	-0.0918
4	3	3.0982	3.0558	3.077	-0.077
5	4	4.064	4.0622	4.0631	-0.0631
6	5	5.0996	5.0808	5.0902	-0.0902
7	6	6.0568	6.095	6.0759	-0.0759
8	7	7.1778	7.0054	7.0916	-0.0916
Jumlah	28	28.5528	28.3018		-0.4273
Rata-rata	3.5	3.5691	3.537725		-0.0534

Tabel 4.8 Data Kalibrasi pada Sensor MPX5050GP (B)

No.	Pembacaan Standar, Xi (Psi)	X_i^2	$Y_i - Y'$	$(Y_i - Y')^2$	$X_i Y_i$
1	0	0	0.0534125	0.002852895	0
2	1	1	0.1157125	0.013389383	0.0623
3	2	4	-0.038388	0.0014736	-0.1836
4	3	9	-0.023588	0.00055637	-0.231
5	4	16	-0.009688	9.38477E-05	-0.2524
6	5	25	-0.036787	0.00135332	-0.451
7	6	36	-0.022487	0.000505688	-0.4554
8	7	49	-0.038187	0.001458285	-0.6412
Jumlah	28	140		0.021683389	-2.1523
Rata-rata	3.5	17.5		0.002710424	-0.269

Tabel 4.9 Data Kalibrasi pada Sensor MPX5050GP (C)

No.	Pembacaan Standar, Xi (Psi)	Y Regresi	$Y_i - Y$ Regresi	$(Y_i - Y \text{ Regresi})^2$
1	0	0.0013	-0.0013	0.00000169
2	1	-0.0143	0.0766	0.00586756
3	2	-0.0299	-0.0619	0.00383161
4	3	-0.0455	-0.0315	0.00099225
5	4	-0.0611	-0.002	0.000004
6	5	-0.0767	-0.0135	0.00018225
7	6	-0.0923	0.0164	0.00026896
8	7	-0.1079	0.0163	0.00026569
Jumlah	28	-0.4264	-0.0009	0.01141401
Rata-rata	3.5	-0.0533	-0.0001125	0.001426751

a. Nilai Ketidakpastian *Type A*:

Berdasarkan persamaan 2.7 didapatkan hasil standar deviasi sebagai berikut:

$$\sigma \text{ (Standar Deviasi)} = \sqrt{\frac{\sum(y_i - \bar{y})^2}{n-1}}$$

$$\sigma = 0.04$$

Sehingga nilai ketidakpastian hasil pengukuran:

$$U_{a1} = \frac{\sigma}{\sqrt{n}}$$

$$U_{a1} = \frac{0.04}{\sqrt{8}} = 0.014$$

Sedangkan nilai ketidakpastian regresi $U_{a2} = \sqrt{\frac{SSR}{n-2}}$

Dimana:

$$Y_{reg} = a + (b \cdot xi)$$

$$a = \frac{(\sum x_i^2)(\sum y_i) - (\sum xi)(\sum xi \cdot y_i)}{n \cdot \sum xi^2 - (\sum xi)^2}$$

$$a = \frac{(140 \times (-0.4273)) - (28 \times (-2.1523))}{(8 \times 140) - 784} = \frac{0.4424}{336} = 0.0013$$

$$b = \frac{n \cdot \sum x_i y_i - \sum xi \sum y_i}{n \cdot \sum xi^2 - (\sum xi)^2}$$

$$b = \frac{(8 \times (-2.1523)) - (28 \times (-0.4273))}{(8 \times 140) - 784} = \frac{-5.254}{336} = -0.0156$$

$$SSR \text{ (Sum Square Residual)} = \sum(y_i - Y_{reg})^2 = 0,0114101$$

Sehingga menghasilkan nilai $U_{a2} =$

$$U_{a2} = \sqrt{\frac{SSR}{n-2}} = \sqrt{\frac{0,0114101}{8-2}} = 0.0436$$

b. Nilai Ketidakpastian *Type B*:

Pada tipe ini terdapat 2 parameter ketidakpastian, yaitu ketidakpastian Resolusi (U_{b1}) dan Ketidakpastian alat standar *pressure gauge* (U_{b2}). Dengan perhitungan sebagai berikut:

$$U_{b1} = \frac{\frac{1}{2} \times \text{Resolusi}}{\sqrt{3}} = \frac{\frac{1}{2} \times 0.01}{\sqrt{3}} = 0.0029$$

$$U_{b2} = \frac{a}{k},$$

nilai a (ketidakpastian sertifikat kalibrasi) dianggap mendekati 0, dan nilai faktor cakupan dianggap 2,0. Sehingga hasil : $U_{b2} = 0$

c. Nilai Ketidakpastian Kombinasi U_c :

Berdasarkan persamaan 2.12 didapatkan hasil U_c sebagai berikut:

$$U_c = \sqrt{U_{a1}^2 + U_{a2}^2 + U_{b1}^2 + U_{b2}^2}$$

$$U_c = \sqrt{0.014^2 + 0.0436^2 + 0.0029^2 + 0^2}$$

$$U_c = 0.046$$

Dengan kondisi V atau derajat kebebasan dari kedua tipe ketidakpastian, sebagai berikut :

$V = n-1$, sehingga :

$V_1 = 7$; $V_2 = 7$; $V_3 = \infty$; $V_4 = 60$ (berdasarkan table *T-Student*)

Dengan nilai V_{eff} (Nilai derajat kebebasan efektif) sebagai berikut :

$$V_{eff} = \frac{(U_c)^4}{\sum (U_i)^4 / V_i}$$

$$V_{eff} = \frac{(0.046)^4}{(0.014)^4 / 7 + (0.0436)^4 / 7 + (0.0029)^4 / \infty + (0.00)^4 / 60}$$

$V_{eff} = 8.58$, sehingga jika dibulatkan menjadi 9, dimana pada table *T-student* menghasilkan nilai k (faktor koreksi) sebesar 2.262.

Oleh karena itu, hasil nilai ketidakpastian diperluas sebesar :

$$U_{exp} = k \times U_c$$

$$U_{exp} = 2.262 \times 0.046 = 0.104$$

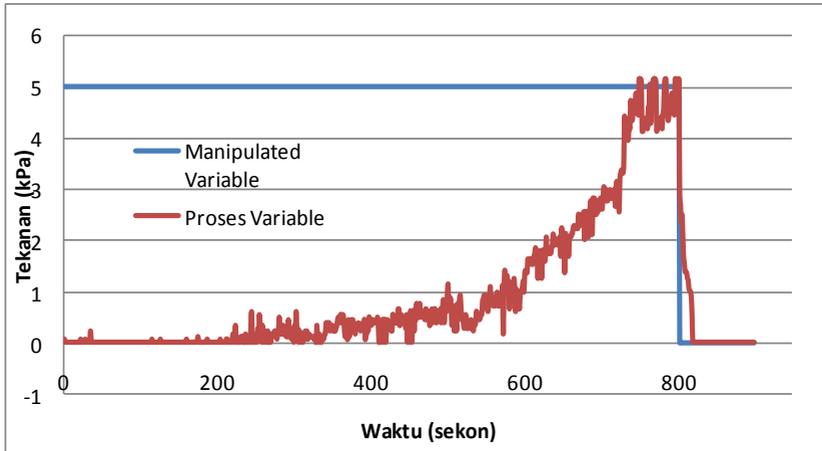
Sehingga berdasarkan perhitungan ketidakpastian diperluas diatas menghasilkan nilai ketidakpastian alat sebesar ± 0.104 dengan tingkat kepercayaan 95% dari tabel *T-Student*. Nilai ketidakpastian tersebut akan menjadi acuan pembacaan alat ukur selama alat ukur tersebut digunakan. Dari nilai ketidakpastian diperluas tersebut, didapatkan bahwa alat masih layak digunakan karena tidak melebihi nilai 5% dari nilai rata-rata pembacaan alat dan tidak melebihi nilai 3σ .

4.3 Pengujian Respon Sistem Pengendalian

Berikut merupakan respon sistem pengendalian tekanan pada *miniplant* labu pemanas terhadap waktu

A. Open Loop

Sistem control dimana keluaran tidak memberikan efek terhadap besaran masukan. Pengendalian tekanan hanya berdasarkan waktu dimana semua kontrol mengacu pada waktu yang ditentukan, disinilah *open loop* bekerja. Sensor MPX5050GP mendeteksi adanya tekanan pada aliran uap dalam satuan kPa yang nantinya diharapkan dapat menggerakkan motor servo untuk membuka *valve*.

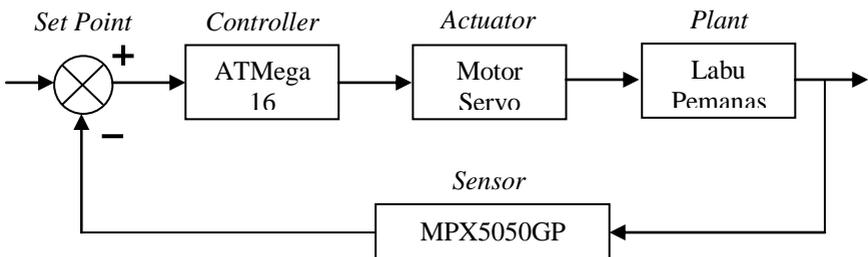


Gambar 4.5 Grafik *Open Loop* Sistem Pengendalian Tekanan

Gambar 4.5 merupakan grafik *open loop* sistem pengendalian tekanan, dimana tekanan mulai terbaca sebesar 0.06 kPa pada saat suhu mencapai 57.37°C .

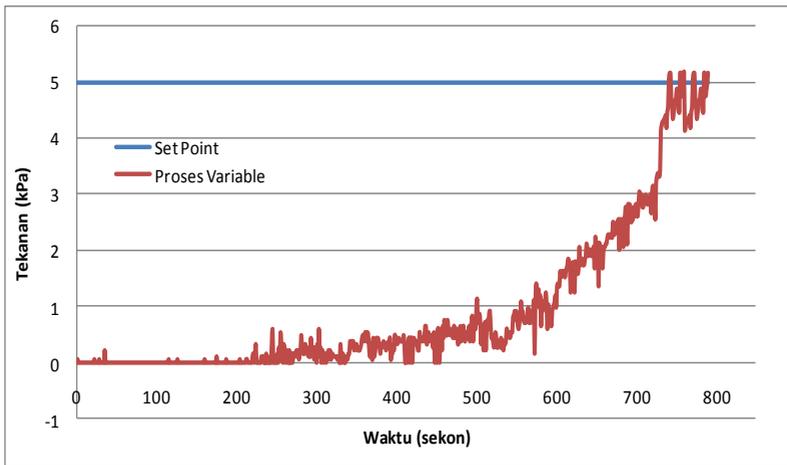
B. *Close Loop*

Sistem control dimana besaran keluaran memberikan efek terhadap besaran masukan sehingga besaran yang dikontrol dapat dibandingkan terhadap nilai yang diinginkan.



Gambar 4.6 Diagram Blok Sistem Pengendalian Tekanan

Gambar 4.6 merupakan gambar diagram blok sistem pengendalian tekanan pada labu pemanas yang terdiri dari mikrokontroler ATmega 16 sebagai *controller*, motor servo sebagai *actuator*, labu pemanas sebagai tempat pemrosesan, dan sensor MPX5050GP sebagai sensor untuk sensing tekanan didalam labu pemanas (dalam satuan KPa). Nilai tekanan yang dibaca oleh sensor akan ditampilkan ke *display* LCD (*Liquid Crystal Display*) 4x20 dalam satuan KPa.



Gambar 4.7 Grafik *Close Loop* Sistem Pengendalian Tekanan

Gambar 4.7 merupakan grafik *close loop* sistem pengendalian tekanan, dimana tekanan mencapai *set point* 5 kPa pada waktu 773 detik.

1. Spesifikasi respon
 - a. *Error Steady State*

$$E_{ss} = 0.04$$

- b. *Maximum Overshoot*

Nilai puncak maksimum dari kurva tanggapan sistem.

$$M_p = \frac{C(tp) - C(\infty)}{C(\infty)} \times 100\%$$

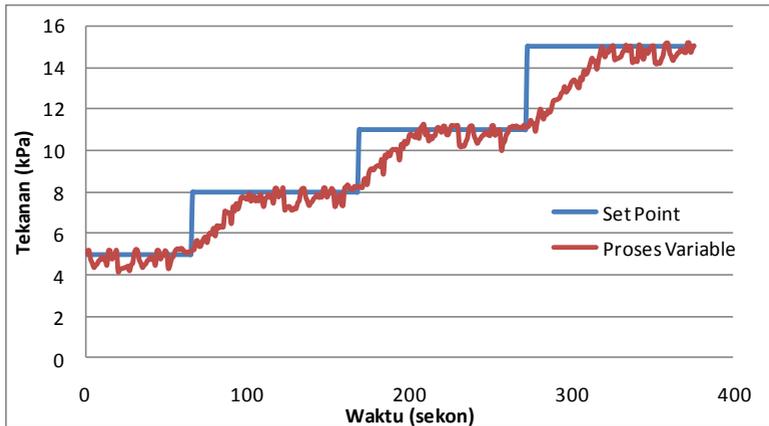
$$= \frac{5.09 - 5}{5} = 1.8\%$$

c. *Settling Time*

Pada pengendalian tekanan ini tidak memiliki *settling time* karena sistem bekerja secara *on off*.

C. *Tracking Set Point*

Tracking setpoint dilakukan dengan cara mengubah nilai *setpoint*. Hasil dari *Tracking setpoint* ditunjukkan oleh gambar 4.8 dimana sistem kontrol yang telah dibuat dapat mengejar nilai *setpoint* ketika diberikan perubahan nilai terhadap nilai *setpoint*.



Gambar 4.8 Grafik Respon *Tracking Set Point* Sistem Pengendalian Tekanan

Gambar 4.8 merupakan grafik respon *tracking set point* sistem pengendalian tekanan, dimana *set point* ditambah sebesar 3 kPa dari *set point* awal yaitu 5 kPa. Ketika *set point* ditambah menjadi 15 kPa memerlukan waktu 310 detik.

4.4 Pembahasan

Tugas akhir yang berjudul sistem pengendalian tekanan pada *miniplant* labu pemanas ini bertujuan untuk mengetahui keadaan tekanan di dalam labu pemanas melalui display pada LCD. Menggunakan sensor tekanan MPX5050GP sebagai alat ukur tekanan. Sebelum sensor ini digunakan perlu dilakukan kalibrasi untuk mengetahui performansi dari sensor tersebut. Kalibrasi sensor termokopel ini menggunakan kompresor. Dilakukan pada *range* 0 Psi hingga 7 Psi. Setelah dilakukan pengujian sensor yaitu dilakukan perhitungan kalibrasi. Dari tabel 4.2 dan tabel 4.4 didapatkan hasil pengukuran. Dari pembacaan tersebut dapat dicari nilai ketidakpastian pengukuran tekanan dengan hasil $U_{a1} = 0.014$ $U_{a2} = 0.0436$, $U_{b1} = 0.0029$, $U_{b2} = 0$ $U_c = 0.046$. Sehingga berdasarkan perhitungan ketidakpastian diperluas tersebut menghasilkan nilai U_{expand} sebesar ± 0.104 dengan tingkat kepercayaan 95% dari tabel *T-Student*. Hasil dari perhitungan ketidakpastian tersebut akan menjadi acuan dari sensor tekanan MPX5050GP yang akan digunakan. Sensor tersebut memiliki karakteristik statik diantaranya resolusi sebesar 0.01, sensitivitas 93.238488 mV/kPa dan akurasi sebesar 82.5%. Dari pembacaan MPX5050GP ini selanjutnya akan diproses oleh mikrokontroler yang nantinya akan memberikan sebuah sinyal kontrol terhadap aktuator yang berupa motor servo untuk mengejar *setpoint* yang bernilai 5 kPa. Ketika saklar pada labu pemanas mulai dinyalakan mikrokontroler akan menyalakan elemen pemanas listrik dan menggerakkan motor servo untuk menutup *valve*. Selama proses pemanasan dilakukan, maka tekanan di dalam labu pemanas semakin bertambah dan ketika tekanan telah mencapai *setpoint* maka mikrokontroler akan menggerakkan motor servo untuk membuka *valve* sehingga tekanan yang ada di dalam labu pemanas selalu sesuai dengan *setpoint*.

(Halaman ini sengaja dikosongkan)

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian sistem pengendalian tekanan pada aliran uap, dapat disimpulkan yaitu:

- a. Telah dibuat rancang bangun sistem pengendalian tekanan pada aliran uap
- b. Pada *miniplant* pengendalian tekanan didapatkan *error steady state* sebesar 0.04 dan *maximum overshoot* sebesar 1.8%. Pada pengendalian tekanan ini tidak memiliki *settling time* karena sistem bekerja secara *on off*.

5.2 Saran

Adapun saran untuk penelitian sistem pengendalian tekanan pada aliran uap adalah sebagai berikut:

- a. Alat ini dapat disempurnakan dengan menambahkan *Safety Instrumented System (SIS)* agar *miniplant* lebih terproteksi keamanannya karena *miniplant* ini merupakan *miniplant* dengan variabel tekanan.
- b. Dibutuhkan labu didih yang lebih kuat dari material yang digunakan pada penelitian ini jika ingin meningkatkan kinerja dan memperoleh hasil yang maksimal dari *miniplant* labu pemanas.

DAFTAR PUSTAKA

- [1] Febriantara. 2008. “*Session 3 Steam Table, Steam Quality & Steam Properties*”.
- [2] Dwi Via. 2016. “Prosedur Penggunaan Alat-alat Laboratorium”.
- [3] Arindya, ST, MT. 2014. “Instrumentasi dan Kontrol Proses”. Graha Ilmu.
- [4] Rizky Achmad. 2014. “Sistema Mikrokontroler Dasar”. Malang
- [5] Robotarduino. 2016. “Mengenal Motor Servo”. Surabaya, Macammacamservo.
- [6] Ismail Denta. 2012. “Rancang Bangun Portabel Tensimeter dan Elektrokardiograf Berbasis Mikrokontroler Arduino”. Surabaya, Jurusan Fisika ITS.
- [7] syahrul Citra. 2011. “*Water Flow Sensor*”. Universitas Sumatera Utara. Sumatera.
- [8] Elektronika Dasar. 2012. “*LCD (Liquid Cristal Display)*”. Komponen, Teori Elektronika

LAMPIRAN A

(LISTING PROGRAM PADA CODE VISION AVR)

```

/*****
This program was created by the
CodeWizardAVR V3.12 Advanced
Automatic Program Generator
© Copyright 1998-2014 Pavel Haiduc, HP InfoTech s.r.l.
http://www.hpinfotech.com

Project :
Version :
Date    : 29/05/2017
Author  :
Company :
Comments:

Chip type      : ATmega16
Program type   : Application
AVR Core Clock frequency: 11,059200 MHz
Memory model   : Small
External RAM size : 0
Data Stack size : 256
*****/

#include <mega16.h>

#include <delay.h>

#include <stdio.h>

#include <stdlib.h>

#define servo   OCR1A      //memberi nama servo pada OCR1A, OCR1A
--> PORTD.5
#define relay   PORTD.6    //memberi nama relay pada PORTD.6
#define tombol1 PINB.0     //memberi nama PINB.0 dengan tombol1
#define tombol2 PINB.1     //memberi nama PINB.1 dengan tombol2
#define tombol3 PINB.2     //memberi nama PINB.2 dengan tombol3
#define tombol4 PINB.3     //memberi nama PINB.3 dengan tombol4

// Alphanumeric LCD functions
#include <alcd.h>

void lcd_3digit(int nilai, int x, int y);
void lcd_4digit(int nilai, int x, int y);
void baca_suhu();
```

```

void baca_tekanan();
void baca_rpm();

//deklarasi variable untuk keperluan interrupt
int frekuensi,i;
float rpm;

// Declare your global variables here
int buka = 3150;
int tutup = 1750;
unsigned data_suhu;
float nilai_suhu = 0;
int setpoint_suhu = 80;
char lcd_buffer[33];
unsigned int adc_tekanan;
float nilai_tegangan = 0, nilai_tekanan = 0, nilai_tegangan_TK =
0;
int setpoint_tekanan = 0;

int flag_suhu = 0, flag_tekanan = 0;
int flag_atur_setpoint = 0, flag_mulai_proses = 0;
int menu = 0, set = 0;

// External Interrupt 0 service routine
interrupt [EXT_INT0] void ext_int0_isr(void)
{
// Place your code here
    frekuensi++; //apabila terdeteksi eksternal interrupt nilai
frekuensi di increment
}

// Timer 0 overflow interrupt service routine
interrupt [TIM0_OVF] void timer0_ovf_isr(void)
{
// Reinitialize Timer 0 value
TCNT0=0x94;
// Place your code here
    i++; //periode timer0 = 10ms,
sehingga perlu hitungan 100 kali increment untuk mendapatkan
periode 1 detik
    if(i>=100)
    {
        rpm=(float)frekuensi*60/6; //formula perhitungan
kecepatan
        frekuensi = 0;
        i = 0;
    }
}

```

```

// Voltage Reference: AVCC pin
#define ADC_VREF_TYPE ((0<<REFS1) | (1<<REFS0) | (0<<ADLAR))

// Read the AD conversion result
unsigned int read_adc(unsigned char adc_input)
{
    ADMUX=adc_input | ADC_VREF_TYPE;
    // Delay needed for the stabilization of the ADC input voltage
    delay_us(10);
    // Start the AD conversion
    ADCSRA|=(1<<ADSC);
    // Wait for the AD conversion to complete
    while ((ADCSRA & (1<<ADIF))==0);
    ADCSRA|=(1<<ADIF);
    return ADCW;
}

// SPI functions
#include <spi.h>

void main(void)
{
    // Declare your local variables here

    // Input/Output Ports initialization
    // Port A initialization
    // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In
    Bit1=In Bit0=In
    DDRA=(0<<DDA7) | (0<<DDA6) | (0<<DDA5) | (0<<DDA4) | (0<<DDA3) |
    (0<<DDA2) | (0<<DDA1) | (0<<DDA0);
    // State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
    PORTA=(0<<PORTA7) | (0<<PORTA6) | (0<<PORTA5) | (0<<PORTA4) |
    (0<<PORTA3) | (0<<PORTA2) | (0<<PORTA1) | (0<<PORTA0);

    // Port B initialization
    // Function: Bit7=Out Bit6=In Bit5=Out Bit4=Out Bit3=In Bit2=In
    Bit1=In Bit0=In
    DDRB=(1<<DDB7) | (0<<DDB6) | (1<<DDB5) | (1<<DDB4) | (0<<DDB3) |
    (0<<DDB2) | (0<<DDB1) | (0<<DDB0);
    // State: Bit7=0 Bit6=T Bit5=0 Bit4=0 Bit3=P Bit2=P Bit1=P Bit0=P
    PORTB=(0<<PORTB7) | (0<<PORTB6) | (0<<PORTB5) | (0<<PORTB4) |
    (1<<PORTB3) | (1<<PORTB2) | (1<<PORTB1) | (1<<PORTB0);

    // Port C initialization
    // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In
    Bit1=In Bit0=In
    DDRC=(0<<DDC7) | (0<<DDC6) | (0<<DDC5) | (0<<DDC4) | (0<<DDC3) |
    (0<<DDC2) | (0<<DDC1) | (0<<DDC0);
    // State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T

```

```

PORTC=(0<<PORTC7) | (0<<PORTC6) | (0<<PORTC5) | (0<<PORTC4) |
(0<<PORTC3) | (0<<PORTC2) | (0<<PORTC1) | (0<<PORTC0);

// Port D initialization
// Function: Bit7=In Bit6=Out Bit5=Out Bit4=Out Bit3=In Bit2=In
Bit1=In Bit0=In
DDRD=(0<<DDD7) | (1<<DDD6) | (1<<DDD5) | (1<<DDD4) | (0<<DDD3) |
(0<<DDD2) | (0<<DDD1) | (0<<DDD0);
// State: Bit7=T Bit6=0 Bit5=0 Bit4=0 Bit3=T Bit2=T Bit1=T Bit0=T
PORTD=(0<<PORTD7) | (0<<PORTD6) | (0<<PORTD5) | (0<<PORTD4) |
(0<<PORTD3) | (0<<PORTD2) | (0<<PORTD1) | (0<<PORTD0);

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: 10,800 kHz
// Mode: Normal top=0xFF
// OC0 output: Disconnected
// Timer Period: 10 ms
TCCR0=(0<<WGM00) | (0<<COM01) | (0<<COM00) | (0<<WGM01) |
(1<<CS02) | (0<<CS01) | (1<<CS00);
TCNT0=0x94;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: 1382,400 kHz
// Mode: Fast PWM top=ICR1
// OC1A output: Non-Inverted PWM
// OC1B output: Disconnected
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer Period: 20 ms
// Output Pulse(s):
// OC1A Period: 20 ms Width: 0 us
// Timer1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=(1<<COM1A1) | (0<<COM1A0) | (0<<COM1B1) | (0<<COM1B0) |
(1<<WGM11) | (0<<WGM10);
TCCR1B=(0<<ICNC1) | (0<<ICES1) | (1<<WGM13) | (1<<WGM12) |
(0<<CS12) | (1<<CS11) | (0<<CS10);
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x6B;
ICR1L=0xFF;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;

```

```

OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer2 Stopped
// Mode: Normal top=0xFF
// OC2 output: Disconnected
ASSR=0<<AS2;
TCCR2=(0<<PWM2) | (0<<COM21) | (0<<COM20) | (0<<CTC2) | (0<<CS22)
| (0<<CS21) | (0<<CS20);
TCNT2=0x00;
OCR2=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=(0<<OCIE2) | (0<<TOIE2) | (0<<TICIE1) | (0<<OCIE1A) |
(0<<OCIE1B) | (0<<TOIE1) | (0<<OCIE0) | (1<<TOIE0);

// External Interrupt(s) initialization
// INT0: On
// INT0 Mode: Any change
// INT1: Off
// INT2: Off
GICR|=(0<<INT1) | (1<<INT0) | (0<<INT2);
MCUCR=(0<<ISC11) | (0<<ISC10) | (0<<ISC01) | (1<<ISC00);
MCUCSR=(0<<ISC2);
GIFR=(0<<INTF1) | (1<<INTF0) | (0<<INTF2);

// USART initialization
// USART disabled
UCSRB=(0<<RXCIEN) | (0<<TXCIEN) | (0<<UDRIEN) | (0<<RXDEN) | (0<<TXDEN)
| (0<<UCSZ2) | (0<<RXB8) | (0<<TXB8);

// Analog Comparator initialization
// Analog Comparator: Off
// The Analog Comparator's positive input is
// connected to the AIN0 pin
// The Analog Comparator's negative input is
// connected to the AIN1 pin
ACSR=(1<<ACD) | (0<<ACBG) | (0<<ACO) | (0<<ACI) | (0<<ACIE) |
(0<<ACIC) | (0<<ACIS1) | (0<<ACIS0);

// ADC initialization
// ADC Clock frequency: 86,400 kHz
// ADC Voltage Reference: AVCC pin
// ADC Auto Trigger Source: ADC Stopped
ADMUX=ADC_VREF_TYPE;
ADCSRA=(1<<ADEN) | (0<<ADSC) | (0<<ADATE) | (0<<ADIF) | (0<<ADIE)
| (1<<ADPS2) | (1<<ADPS1) | (1<<ADPS0);
SFIOR=(0<<ADTS2) | (0<<ADTS1) | (0<<ADTS0);

```

```

// SPI initialization
// SPI Type: Master
// SPI Clock Rate: 2764,800 kHz
// SPI Clock Phase: Cycle Start
// SPI Clock Polarity: Low
// SPI Data Order: MSB First
SPCR=(0<<SPIE) | (1<<SPE) | (0<<DORD) | (1<<MSTR) | (0<<CPOL) |
(0<<CPHA) | (0<<SPR1) | (0<<SPR0);
SPSR=(0<<SPI2X);

// TWI initialization
// TWI disabled
TWCR=(0<<TWEA) | (0<<TWSTA) | (0<<TWSTO) | (0<<TWEN) | (0<<TWIE);

// Alphanumeric LCD initialization
// Connections are specified in the
// Project|Configure|C Compiler|Libraries|Alphanumeric LCD menu:
// RS - PORTC Bit 0
// RD - PORTC Bit 1
// EN - PORTC Bit 2
// D4 - PORTC Bit 4
// D5 - PORTC Bit 5
// D6 - PORTC Bit 6
// D7 - PORTC Bit 7
// Characters/line: 20
lcd_init(20);

// Global enable interrupts
#asm("sei")
relay = 0; //matikan kompor dulu, karena relaynya aktifnya kalau
diberi nilai 0 (kompor normally close)
servo = tutup; //valve tertutup

//tampilan awal
lcd_gotoxy(5,0); //text dimulai pada koordinat 0,0
lcd_putsf("Bismillah");
lcd_gotoxy(3,1);
lcd_putsf("Heating Flask");
lcd_gotoxy(3,2);
lcd_putsf("Semoga Sukses");
lcd_gotoxy(4,3);
lcd_putsf("Aamiin...!!!");
delay_ms(2000);
lcd_clear();

while (1)
{
// Place your code here

```

```

servo = buka; //valve tertutup
relay = 0;
//lcd_clear();
lcd_gotoxy(0,0);
lcd_putsf("      Menu      ");
lcd_gotoxy(1,1);
lcd_putsf("1.Atur Setpoint ");
lcd_gotoxy(1,2);
lcd_putsf("2.Reset Setpoint");
lcd_gotoxy(1,3);
lcd_putsf("3.Mulai Proses      ");

if(tombol4 == 0) //jika tombol tambah ditekan
{
    menu++;          //+1
    delay_ms(500); //didelay setiap penambahannya
    if(menu > 2)
    {
        menu = 0; //menu kembali ke awal jika nilai menu > 2
    }
}

if(tombol3 == 0) //jika tombol kurang ditekan
{
    menu--;          //-1
    delay_ms(500);
    if(menu < 0)
    {
        menu = 2;
    }
}

if(menu == 0)
{
    lcd_gotoxy(0,1);
    lcd_putsf(">");
    lcd_gotoxy(0,2);
    lcd_putsf(" ");
    lcd_gotoxy(0,3);
    lcd_putsf(" ");
}

if(menu == 1)
{
    lcd_gotoxy(0,1);
    lcd_putsf(" ");
    lcd_gotoxy(0,2);
    lcd_putsf(">");
    lcd_gotoxy(0,3);
}

```

```

        lcd_putsf(" ");
    }

    if(menu == 2)
    {
        lcd_gotoxy(0,1);
        lcd_putsf(" ");
        lcd_gotoxy(0,2);
        lcd_putsf(" ");
        lcd_gotoxy(0,3);
        lcd_putsf(">");
    }

    if(tombol1 == 0 && menu == 0)
    {
        delay_ms(200);
        flag_atur_setpoint = 1;
    }

    if(tombol1 == 0 && menu == 1)
    {
        lcd_clear();
        lcd_gotoxy(2,1);
        lcd_putsf("Setpoint Direset");
        setpoint_suhu = 80;
        setpoint_tekanan = 0;
        delay_ms(2000);
    }

    if(tombol1 == 0 && menu == 2 && setpoint_suhu > 80 &&
setpoint_tekanan > 0)
    {
        flag_mulai_proses = 1;
    }

    if(tombol1 == 0 && menu == 2 && setpoint_suhu == 80 &&
setpoint_tekanan == 0)
    {
        lcd_clear();
        lcd_gotoxy(3,1);
        lcd_putsf("Atur Setpoint");
        lcd_gotoxy(7,2);
        lcd_putsf("Dahulu");
        delay_ms(2000);
    }

    //program atur setpoint
    while(flag_atur_setpoint == 1)
    {

```

```

lcd_clear();
lcd_gotoxy(1,0);
lcd_putsf("1.Atur Suhu");
lcd_gotoxy(1,2);
lcd_putsf("2.Atur Tekanan");
lcd_gotoxy(13,3);
lcd_putsf("Selesai");
itoa(setpoint_suhu,lcd_buffer);
lcd_4digit(setpoint_suhu,1,1);
itoa(setpoint_tekanan,lcd_buffer);
lcd_3digit(setpoint_tekanan,1,3);
if(tombol4 == 0)
{
    set++;
    delay_ms(200);
    if(set > 2)
    {
        set = 0;
    }
}
if(tombol3 == 0)
{
    set--;
    delay_ms(200);
    if(set < 0)
    {
        set = 2;
    }
}
if(set == 0)
{
    lcd_gotoxy(0,0);
    lcd_putsf(">");
    lcd_gotoxy(0,2);
    lcd_putsf(" ");
    lcd_gotoxy(12,3);
    lcd_putsf(" ");
}
if(set == 1)
{
    lcd_gotoxy(0,0);
    lcd_putsf(" ");
    lcd_gotoxy(0,2);
    lcd_putsf(">");
    lcd_gotoxy(12,3);
    lcd_putsf(" ");
}
if(set == 2)
{

```

```

        lcd_gotoxy(0,0);
        lcd_putsf(" ");
        lcd_gotoxy(0,2);
        lcd_putsf(" ");
        lcd_gotoxy(12,3);
        lcd_putsf(">");
    }
    if(tombol1 == 0 && set == 0)
    {
        delay_ms(200);
        flag_suhu = 1;
    }
    if(tombol1 == 0 && set == 1)
    {
        delay_ms(200);
        flag_tekanan = 1;
    }
    if(tombol1 == 0 && set == 2)
    {
        delay_ms(200);
        flag_atur_setpoint = 0;
    }
    if(tombol2 == 0)
    {
        delay_ms(200);
        flag_atur_setpoint = 0;
        setpoint_suhu = 80;
        setpoint_tekanan = 0;
    }
    while(flag_suhu == 1)
    {
        lcd_clear();
        lcd_gotoxy(0,0);
        lcd_putsf(">");
        lcd_gotoxy(1,0);
        lcd_putsf("1.Atur Suhu<");
        lcd_gotoxy(1,2);
        lcd_putsf("2.Atur Tekanan ");
        if(tombol3 == 0)
        {
            setpoint_suhu++;
            delay_ms(200);
            if(setpoint_suhu > 1000)
            {
                setpoint_suhu = 1000;
            }
        }
    }
    if(tombol4 == 0)
    {

```

```

        setpoint_suhu--;
        delay_ms(200);
        if(setpoint_suhu < 80)
        {
            setpoint_suhu = 80;
        }
    }
    itoa(setpoint_suhu,lcd_buffer);
    lcd_4digit(setpoint_suhu,1,1);
    itoa(setpoint_tekanan,lcd_buffer);
    lcd_3digit(setpoint_tekanan,1,3);
    if(tombol1 == 0 && setpoint_suhu > 80)
    {
        delay_ms(200);
        flag_suhu = 0;
    }
    if(tombol2 == 0)
    {
        delay_ms(200);
        flag_suhu = 0;
        setpoint_suhu = 80;
    }
    delay_ms(50);
}
while(flag_tekanan == 1)
{
    lcd_clear();
    lcd_gotoxy(1,0);
    lcd_putsf("1.Atur Suhu");
    lcd_gotoxy(0,2);
    lcd_putsf(">");
    lcd_gotoxy(1,2);
    lcd_putsf("2.Atur Tekanan<");
    if(tombol3 == 0)
    {
        setpoint_tekanan++;
        delay_ms(200);
        if(setpoint_tekanan > 50)
        {
            setpoint_tekanan = 50;
        }
    }
}
if(tombol4 == 0)
{
    setpoint_tekanan--;
    delay_ms(200);
    if(setpoint_tekanan < 0)
    {
        setpoint_tekanan = 0;
    }
}

```

```

    }
}
itoa(setpoint_suhu,lcd_buffer);
lcd_4digit(setpoint_suhu,1,1);
itoa(setpoint_tekanan,lcd_buffer);
lcd_3digit(setpoint_tekanan,1,3);
if(tombol1 == 0 && setpoint_tekanan > 0)
{
    delay_ms(200);
    flag_tekanan = 0;
}
if(tombol2 == 0)
{
    delay_ms(200);
    flag_tekanan = 0;
    setpoint_tekanan = 0;
}
delay_ms(50);
}
delay_ms(50);
}

//program mulai proses
while(flag_mulai_proses == 1)
{
    if(tombol2 == 0)
    {
        flag_mulai_proses = 0;
    }

    lcd_gotoxy(3,0);
    lcd_putsf("Pembacaan Data");
    baca_suhu();
    baca_tekanan();
    baca_rpm();

    //kontrol servo
    //3000 --> +90 derajat
    //1750 --> 0 derajat
    //600 --> -90 derajat (ketiga data tersebut dari coba-coba

    if(nilai_suhu >= setpoint_suhu)
    {
        flag_suhu = 1;
    }
    else
    {
        relay = 1;    //menyalakan kompor
    }
}

```

```

        if((int)nilai_tekanan >= setpoint_tekanan-3 &&
(int)nilai_tekanan < setpoint_tekanan)
        {
            servo = tutup + 700;
        }
        if(nilai_tekanan >= setpoint_tekanan)
        {
            flag_tekanan = 1;
        }
        else
        {
            servo = tutup; //valve tertutup
        }

        while(flag_suhu == 1)
        {
            lcd_gotoxy(3,0);
            lcd_putsf("Pembacaan Data");
            baca_suhu();
            baca_tekanan();
            baca_rpm();

            relay = 0;    //matikan kompor

            if(nilai_suhu <= setpoint_suhu - 5)
            {
                flag_suhu = 0;
            }
            if((int)nilai_tekanan >= setpoint_tekanan-3 &&
(int)nilai_tekanan < setpoint_tekanan)
            {
                servo = tutup + 700;
            }
            if(nilai_tekanan >= setpoint_tekanan)
            {
                flag_tekanan = 1;
            }
            if(tombol2 == 0)
            {
                flag_mulai_proses = 0;
                flag_suhu = 0;
            }
            else
            servo = tutup;
            while(flag_tekanan == 1)
            {
                lcd_gotoxy(3,0);
                lcd_putsf("Pembacaan Data");
                baca_suhu();

```

```

        baca_tekanan();
        baca_rpm();

        //servo = buka; //valve dibuka
        if((int)nilai_tekanan >= setpoint_tekanan-3 &&
(int)nilai_tekanan < setpoint_tekanan)
        {
            servo = buka - 700;
        }
        if(nilai_tekanan <= setpoint_tekanan - 3)
        {
            flag_tekanan = 0;
        }
        if(tombol2 == 0)
        {
            flag_tekanan = 0;
            flag_mulai_proses = 0;
        }
        if(nilai_tekanan >= setpoint_tekanan)
        {
            servo = buka;
            delay_ms(3000);
        }
    }
}
while(flag_tekanan == 1)
{
    lcd_gotoxy(3,0);
    lcd_putsf("Pembacaan Data");
    baca_suhu();
    baca_tekanan();
    baca_rpm();

    //servo = buka; //valve dibukaa
    relay = 1;

    if(nilai_suhu >= setpoint_suhu)
    {
        flag_suhu = 1;
    }
    if((int)nilai_tekanan >= setpoint_tekanan-3 &&
(int)nilai_tekanan < setpoint_tekanan)
    {
        servo = buka - 700;
    }
    if(nilai_tekanan <= setpoint_tekanan - 3)
    {
        flag_tekanan = 0;
    }
}

```

```

    if(tombol2 == 0)
    {
        flag_tekanan = 0;
        flag_mulai_proses = 0;
    }
    if(nilai_tekanan >= setpoint_tekanan)
    {
        servo = buka;
        delay_ms(3000);
    }
    while(flag_suhu == 1)
    {
        lcd_gotoxy(3,0);
        lcd_putsf("Pembacaan Data");
        baca_suhu();
        baca_tekanan();
        baca_rpm();

        relay = 0; //kompomr dimatikan

        if(nilai_suhu <= setpoint_suhu - 5)
        {
            flag_suhu = 0;
        }
        if(tombol2 == 0)
        {
            flag_suhu = 0;
            flag_mulai_proses = 0;
        }
    }
}
//delay_ms(10);
}

void baca_suhu()
{
    //baca thermocouple type-K dengan MAX6675
    //baca MSB dengan SPI
    PORTB.4 = 0; //baca sensor diaktifkan
    data_suhu = (unsigned) spi(0)<<8; //baca byte MSB
    //baca LSB dengan SPI dan gabungkan dengan MSB
    data_suhu |= spi(0);
    PORTB.4 = 1; //baca sensor dimatikan
    delay_ms(200);
    lcd_clear();
}

```

```

    nilai_suhu = (((float)data_suhu*5000)/4096)/40;    //4096 =
12bit
    ftoa(nilai_suhu,2,lcd_buffer);
    lcd_4digit(nilai_suhu,0,1);
    sprintf(lcd_buffer,"%cC",0xDF);
    lcd_gotoxy(7,1);
    lcd_puts(lcd_buffer);
    //hitung tegangan dalam mV
    nilai_tegangan_TK = (((float)data_suhu*5000)/4096);
    ftoa(nilai_tegangan_TK,2,lcd_buffer);
    lcd_4digit(nilai_tegangan_TK,11,1);
    lcd_gotoxy(18,1);
    lcd_putsf("mV");
}

void baca_tekanan()
{
    //baca tekanan pada sensor MPX5050GP
    adc_tekanan = read_adc(1);
    nilai_tegangan = (float)adc_tekanan*5/1024;
    nilai_tekanan = (((float)nilai_tegangan/5)-0.04)/0.018;
    //nilai_tegangan = (float)adc_tekanan*0.004594330400782013;
    //nilai_tekanan = ((float)((float)nilai_tegangan-0.2)/0.09);
    if(nilai_tekanan < 0) nilai_tekanan = 0;
    ftoa(nilai_tekanan,2,lcd_buffer);
    lcd_3digit(nilai_tekanan,0,2);
    lcd_gotoxy(7,2);
    lcd_putsf("kPa");
    ftoa(nilai_tegangan,2,lcd_buffer);
    lcd_3digit(nilai_tegangan,11,2);
    lcd_gotoxy(16,2);
    lcd_putsf("V");
}

void baca_rpm()
{
    //baca RPM
    ftoa(rpm,0,lcd_buffer);
    lcd_4digit(rpm,0,3);
    lcd_gotoxy(5,3);
    lcd_putsf("RPM");
}

void lcd_3digit(int nilai, int x, int y)
{
    if(nilai >= 100)
    {
        lcd_gotoxy(x,y);
        lcd_puts(lcd_buffer);
    }
}

```

```

    }
    if(nilai < 100 && nilai >= 10)
    {
        lcd_gotoxy(x,y);
        lcd_putsf("0");
        lcd_gotoxy(x+1,y);
        lcd_puts(lcd_buffer);
    }
    if(nilai < 10)
    {
        lcd_gotoxy(x,y);
        lcd_putsf("00");
        lcd_gotoxy(x+2,y);
        lcd_puts(lcd_buffer);
    }
}

void lcd_4digit(int nilai, int x, int y)
{
    if(nilai >= 1000)
    {
        lcd_gotoxy(x,y);
        lcd_puts(lcd_buffer);
    }
    if(nilai < 1000 && nilai >= 100)
    {
        lcd_gotoxy(x,y);
        lcd_putsf("0");
        lcd_gotoxy(x+1,y);
        lcd_puts(lcd_buffer);
    }
    if(nilai < 100 && nilai >= 10)
    {
        lcd_gotoxy(x,y);
        lcd_putsf("00");
        lcd_gotoxy(x+2,y);
        lcd_puts(lcd_buffer);
    }
    if(nilai < 10)
    {
        lcd_gotoxy(x,y);
        lcd_putsf("000");
        lcd_gotoxy(x+3,y);
        lcd_puts(lcd_buffer);
    }
}

```

LAMPIRAN B

(DATA SHEET MINIMUM SYSTEM ATMEGA16)

Features

- High-performance, Low-power Atmel® AVR® 8-bit Microcontroller
- Advanced RISC Architecture
 - 131 Powerful Instructions – Most Single-clock Cycle Execution
 - 32 x 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 16 MIPS Throughput at 16 MHz
 - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory segments
 - 16 Kbytes of In-System Self-programmable Flash program memory
 - 512 Bytes EEPROM
 - 1 Kbyte Internal SRAM
 - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
 - Data retention: 20 years at 85°C/100 years at 25°C¹
 - Optional Boot Code Section with Independent Lock Bits
 - In-System Programming by On-chip Boot Program
 - True Read-While-Write Operation
 - Programming Lock for Software Security
- JTAG (IEEE std. 1149.1 Compliant) Interface
 - Boundary-scan Capabilities According to the JTAG Standard
 - Extensive On-chip Debug Support
 - Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface
- Peripheral Features
 - Two 8-bit Timers/Counters with Separate Prescalers and Compare Modes
 - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Four PWM Channels
 - 8-channel, 10-bit ADC
 - 8 Single-ended Channels
 - 7 Differential Channels in TQFP Package Only
 - 2 Differential Channels with Programmable Gain at 1x, 10x, or 20x
 - Byte-oriented Two-wire Serial Interface
 - Programmable Serial USART
 - Master/Slave SPI Serial Interface
 - Programmable Watchdog Timer with Separate On-chip Oscillator
 - On-chip Analog Comparator
- Special Microcontroller Features
 - Power-on Reset and Programmable Brown-out Detection
 - Internal Calibrated RC Oscillator
 - External and Internal Interrupt Sources
 - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby
- I/O and Packages
 - 32 Programmable I/O Lines
 - 48-pin PDIP, 44-lead TQFP, and 44-pad QFNMLF
- Operating Voltages
 - 2.7V - 5.5V for ATmega16L
 - 4.5V - 5.5V for ATmega16
- Speed Grades
 - 0 - 8 MHz for ATmega16L
 - 0 - 16 MHz for ATmega16
- Power Consumption @ 1 MHz, 3V, and 25°C for ATmega16L
 - Active: 1.1 mA
 - Idle Mode: 0.35 mA
 - Power-down Mode: < 1 µA



8-bit AVR[®]
Microcontroller
with 16K Bytes
In-System
Programmable
Flash

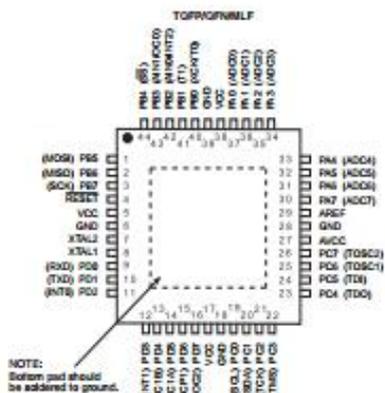
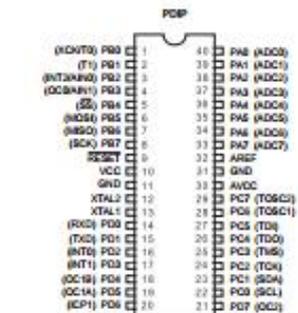
ATmega16
ATmega16L

Rev. 2468T-AVR-07/10



Pin Configurations

Figure 1. Pinout ATmega16



Disclaimer

Typical values contained in this datasheet are based on simulations and characterization of other AVR microcontrollers manufactured on the same process technology. Min and Max values will be available after the device is characterized.

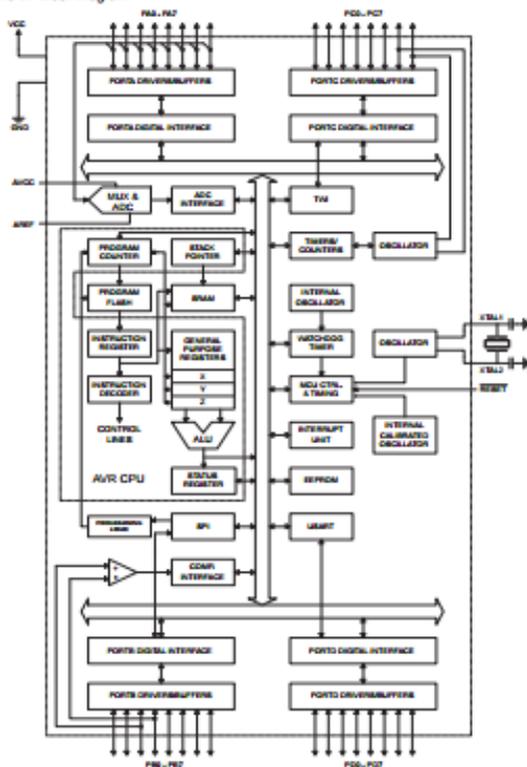


Overview

The ATmega16 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega16 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

Block Diagram

Figure 2. Block Diagram



LAMPIRAN C

(DATA SHEET MPX5050GP)

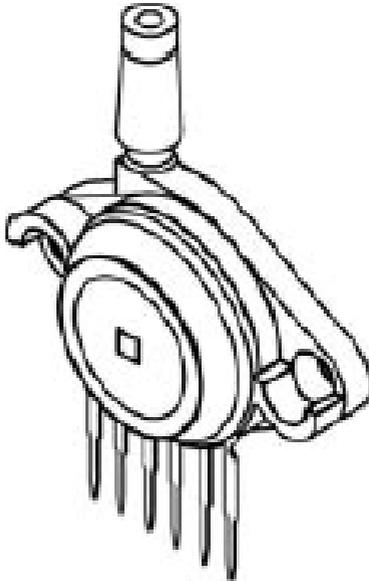
Features

- 2.5% Maximum Error over 0° to 85°C
- Ideally suited for Microprocessor or Microcontroller-Based Systems
- Temperature Compensated Over -40° to +125°C
- Patented Silicon Shear Stress Strain Gauge
- Durable Epoxy Unibody Element
- Easy-to-Use Chip Carrier Option

ORDERING INFORMATION

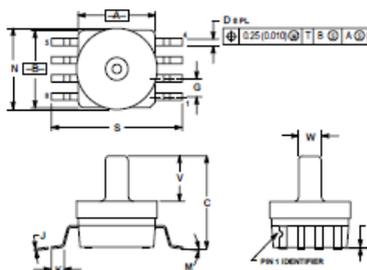
Device Name	Case No.	# of Ports			Pressure Type			Device Marking
		None	Single	Dual	Gauge	Differential	Absolute	
Unibody Package (MPX5050 Series)								
MPX5050D	867	•				•		MPX5050D
MPX5050DP	867C			•		•		MPX5050DP
MPX5050GP	867B		•		•			MPX5050GP
MPX5050GP1	867B		•		•			MPX5050GP
Small Outline Package (MPXV5050 Series)								
MPXV5050GP	1369		•		•			MPXV5050GP
MPXV5050DP	1351			•		•		MPXV5050DP
MPXV5050GC6U	482A		•		•			MPXV5050G
MPXV5050GC6T1	482A		•		•			MPXV5050G
Small Outline Package (Media Resistant Gel) (MPVZ5050 Series)								
MPVZ5050GW7U	1560		•		•			MZ5050GW

Characteristic	Symbol	Min	Typ	Max	Unit
Pressure Range ⁽¹⁾	P _{OP}	0	—	50	kPa
Supply Voltage ⁽²⁾	V _S	4.75	5.0	5.25	Vdc
Supply Current	I _o	—	7.0	10	mAdc
Minimum Pressure Offset ⁽³⁾ @ V _S = 5.0 Volts	V _{off}	0.088	0.2	0.313	Vdc
Full Scale Output ⁽⁴⁾ @ V _S = 5.0 Volts	V _{FSS}	4.587	4.7	4.813	Vdc
Full Scale Span ⁽⁵⁾ @ V _S = 5.0 Volts	V _{FSS}	—	4.5	—	Vdc
Accuracy ⁽⁶⁾	—	—	—	±2.5	%V _{FSS}
Sensitivity	V/P	—	90	—	mV/kPa
Response Time ⁽⁷⁾	t _R	—	1.0	—	ms
Output Source Current at Full Scale Output	I _{o+}	—	0.1	—	mAdc
Warm-Up Time ⁽⁸⁾	—	—	20	—	ms
Offset Stability ⁽⁹⁾	—	—	±0.5	—	%V _{FSS}



MPX5050GP
CASE 867B-04

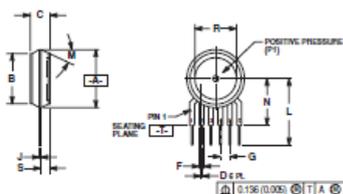
PACKAGE DIMENSIONS



- NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, ASH.
 2. CONTROLLING DIMENSION: INCH.
 3. DIMENSION A AND E DO NOT INCLUDE MOLD PROTECTION.
 4. MAXIMUM MOLD PROTECTION IS 0.008.
 5. ALL VERTICAL SURFACES TYPICAL DRAFT.

DIM	INCHES		MILLIMETERS	
	MIN	MAX	MIN	MAX
A	0.415	0.425	10.54	10.80
B	0.424	0.434	10.76	10.92
C	0.000	0.005	0.00	0.13
D	0.000	0.005	0.00	0.13
E	0.000	0.005	0.00	0.13
F	0.000	0.005	0.00	0.13
G	0.000	0.005	0.00	0.13
H	0.000	0.005	0.00	0.13
J	0.000	0.005	0.00	0.13
K	0.000	0.005	0.00	0.13
M	0.110	0.120	2.79	3.05
N	0.110	0.120	2.79	3.05
S	0.000	0.005	0.00	0.13
T	0.000	0.005	0.00	0.13
W	0.000	0.005	0.00	0.13

CASE 482A-01
ISSUE A
UNIBODY PACKAGE



- NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, ASH.
 2. CONTROLLING DIMENSION: INCH.
 3. DIMENSION A IS INCLUDING OF THE MOLD PROTECTION MOLD PROTECTION IS 0.008 (0.032).

DIM	INCHES		MILLIMETERS	
	MIN	MAX	MIN	MAX
A	0.136	0.141	3.45	3.58
B	0.134	0.139	3.40	3.53
C	0.000	0.005	0.00	0.13
D	0.000	0.005	0.00	0.13
E	0.000	0.005	0.00	0.13
F	0.000	0.005	0.00	0.13
G	0.000	0.005	0.00	0.13
H	0.000	0.005	0.00	0.13
J	0.000	0.005	0.00	0.13
K	0.000	0.005	0.00	0.13
L	0.000	0.005	0.00	0.13
M	0.110	0.120	2.79	3.05
N	0.110	0.120	2.79	3.05
R	0.000	0.005	0.00	0.13
S	0.000	0.005	0.00	0.13

STYLE 1:
PIN 1: VOLT
2: GROUND
3: VOLT
4: VOLT
5: VOLT
6: VOLT

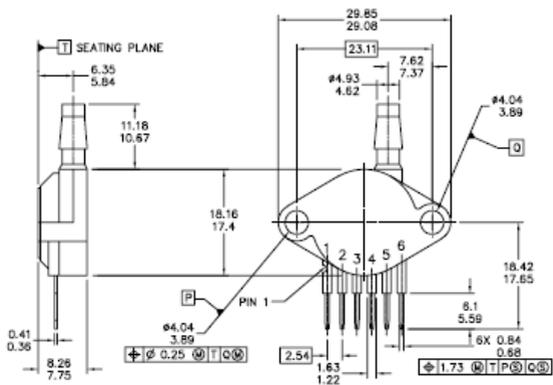
STYLE 2:
PIN 1: OPEN
2: GROUND
3: VOLT
4: VOLT
5: VOLT
6: OPEN

STYLE 3:
PIN 1: OPEN
2: GROUND
3: VOLT
4: VOLT
5: VOLT
6: OPEN

CASE 867-08
ISSUE N
UNIBODY PACKAGE

MPX6060

PACKAGE DIMENSIONS



© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED	MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE
TITLE: SENSOR, 6 LEAD UNBODY CELL, AP & GP 01ASB09087B	DOCUMENT NO: 98ASB42796B	REV: G
	CASE NUMBER: 867B-04	28 JUL 2005
	STANDARD: NON-JEDEC	

PAGE 1 OF 2

CASE 867B-04
ISSUE G
UNIBODY PACKAGE

LAMPIRAN D

(DATA SHEET WATER FLOW SENSOR G1/2)

G 1/2 Water Flow sensor

Introduction

Water flow sensor consists of a plastic valve body, a water rotor, and a hall-effect sensor. When water flows through the rotor, rotor rolls. Its speed changes with different rate of flow. The hall-effect sensor outputs the corresponding pulse signal.

Model: [POW110D3B](#)



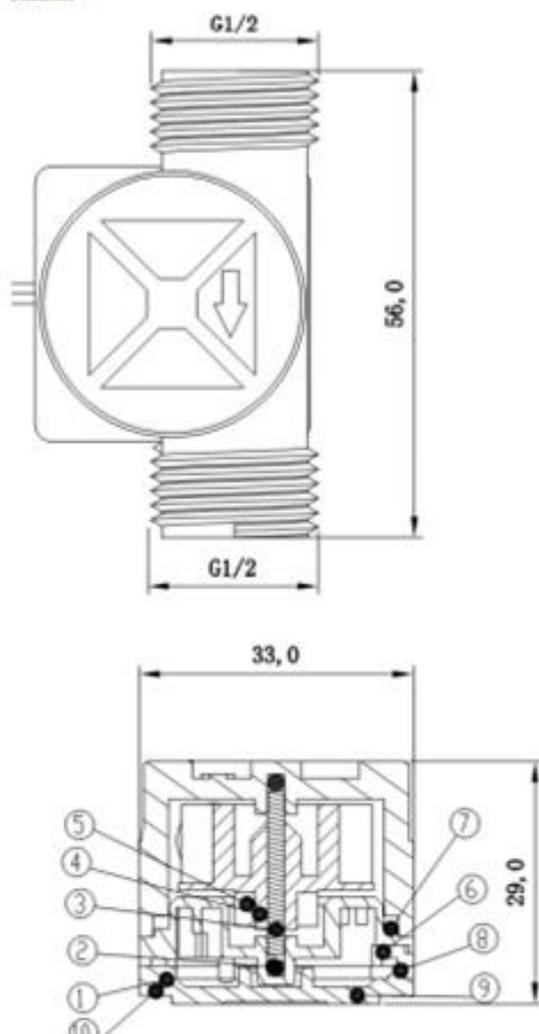
Specification

Working voltage	5V-24V
Maximum current	15 mA (DC 5V)
Weight	43 g
External diameters	20mm
Flow rate range	1~30 L/min
Operating temperature	0°C~80°C
Liquid temperature	<120°C
Operating humidity	35%~90%RH
Operating pressure	under 1.2Mpa
Store temperature	-25°C~+80°C

Mechanic Dimensions

Mechanic Dimensions

Unit:mm



Sensor Components

No.	Name	Quantity	Material	Note
1	Valve body	1	PA66+33%glass fiber	
2	Stainless steel bead	1	Stainless steel SUS304	
3	Axis	1	Stainless steel SUS304	
4	Impeller	1	POM	
5	Ring magnet	1	Ferrite	
6	Middle ring	1	PA66+33%glass fiber	
7	O-seal ring	1	Rubber	
8	Electronic seal ring	1	Rubber	
9	Cover	1	PA66+33%glass fiber	
10	Screw	4	Stainless steel SUS304	3.0*11
11	Cable	1	1007 24AWG	

Usage Example

Note: This example is abstracted from the forum, which was done by Charles Gantt. Thanks for his contribution. Let's see how it works.

Measuring Water Flow rate with Water Flow Sensor

This is part of a project I have been working on and I thought I would share it here since there have been a few reads on how to read water flow rate in liters per hour using the Water Flow Sensor found in the Seeed Studio repo. It uses a simple rotating wheel that pulses a hall effect sensor. By reading these pulses and implementing a little math, we can read the liquids flow rate accurate to within 3%. The threads are simple G1/2 so finding barbed ends will not be that hard.

Hardware Installation

You will need Seeeduino / Arduino, Water Flow Sensor, 10K resistor, a breadboard and some jumper wires.

Wiring up the Water Flow Sensor is pretty simple. There are 3 wires: Black, Red, and Yellow. Black to the Seeeduino's ground pin Red to Seeeduino's 5v pin The yellow wire will need to be connected to a 10k pull up resistor, and then to pin 2 on the Seeeduino.

BIODATA PENULIS



Nama lengkap penulis Azania Arnada Auludyah yang dilahirkan di Gresik pada tanggal 7 April 1996 dari ayah bernama Bambang Agus Hariyanto dan ibu bernama Mushofa. Penulis merupakan anak sulung dari dua bersaudara. Saat ini penulis tinggal di Platinum Residence B-02 Kedayang Gresik. Pada tahun 2008, penulis menyelesaikan pendidikan tingkat dasar di SDN Kawisanyar Gresik. Pada tahun 2011 penulis menyelesaikan pendidikan tingkat menengah pertama di SMP Negeri 1 Gresik. Tahun 2014 berhasil menyelesaikan pendidikan tingkat menengah atas di SMA Muhammadiyah 1 Gresik. Dan pada tahun 2017 ini, penulis mampu menyelesaikan gelar ahli madya di Program Studi DIII-Teknik Instrumentasi, Departemen Teknik Instrumentasi Fakultas Vokasi Institut Teknologi Sepuluh Nopember Surabaya. Penulis berhasil menyelesaikan Tugas Akhir dengan judul “**RANCANG BANGUN SISTEM PENGENDALIAN TEKANAN PADA ALIRAN UAP**”. Bagi pembaca yang memiliki kritik, saran, atau ingin berdiskusi lebih lanjut mengenai Tugas Akhir ini maka dapat menghubungi penulis melalui email azaniaarnada@gmail.com.