

BAB I PENDAHULUAN

1.1 Latar Belakang

Energi merupakan suatu kebutuhan yang sangat penting dewasa ini. Sejalan dengan perkembangan zaman dan kemajuan peradaban manusia, kebutuhan akan energi semakin meningkat dari waktu ke waktu. Dari data *International Energy Agency* (IEA) bahwa permintaan energi rata-rata akan mengalami peningkatan sebesar 1,6 % per tahun. Dari keseluruhan kebutuhan energi dunia tersebut, sekitar 80 % nya dipasok dari bahan bakar fosil, yang mana menurut para ahli minyak bumi bahwasanya bahan bakar fosil diperkirakan akan habis sekitar 30 tahun lagi. Dengan adanya permasalahan ini, sangat diperlukan sumber bahan bakar alternatif sebagai pengganti bahan bakar fosil agar bisa mengatasi hal tersebut.

Bensin dan solar merupakan jenis dari bahan bakar fosil yang paling banyak penggunaannya pada kendaraan bermotor. Menyadari hal ini, telah dilakukan pencarian alternatif untuk mengganti bahan bakar tersebut, salah satunya yaitu bioetanol. Bioetanol merupakan bahan bakar pengganti bensin yang cukup menjanjikan yang didapatkan dari proses fermentasi bahan-bahan nabati yang memiliki kandungan gula. Bahan-bahan tersebut dapat berupa tebu, singkong, kentang, kayu, rumput, dan sebagainya (C. E. Wyman, 1996).

Proses pembuatan bioetanol sendiri diawali dengan *ekstraksi* gula dari bahan-bahan nabati yang terkandung didalamnya. Setelah itu dilakukan fermentasi yang akan menghasilkan bioetanol dengan prosentase 7 % - 10 %. Proses selanjutnya yaitu melakukan distilasi yang menghasilkan alkohol dengan prosentase sekitar 70 % dan teknik *reflux* sehingga menghasilkan konsentrasi 95 %. Untuk meningkatkan kadar sampai 99,5 % biasanya dilakukan dengan penambahan zat pengabsorpsi air. Proses ini memakan waktu 2 – 3 hari. Cara yang lain untuk mengatasi waktu yang lama pada pengabsorpsi air adalah dengan mengatur tekanan sampai mendekati kondisi vakum. Pada kondisi vakum titik uap etanol dan air akan turun, sehingga mempercepat proses distilasi. Namun karena perbedaan titik uap cukup sempit, diperlukan strategi kontrol yang tepat untuk mengatasinya.

Pada skripsi sebelumnya (Akhmad Salmi F., 2014), telah diupayakan sistem kontrol suhu dengan menggunakan kontroler PID. Pada skripsi tersebut hanya satu variabel kontrol saja yang digunakan yaitu suhu, sedangkan parameter tekanan vakum (*vacuum pressure*) dianggap konstan. Hasil yang didapat dari penelitian ini belum memenuhi target yang diharapkan, yaitu diperolehnya *Fuel Grade Ethanol* (Bioetanol dengan kadar lebih dari 99,5 %).

Pada skripsi ini akan diupayakan suatu sistem kontrol yang terpadu antara suhu dan tekanan vakum. Tekanan vakum akan diperlakukan sebagai *disturbance*, yang akan diperhitungkan dalam Algoritma Logika Fuzzy.

Penggunaan Algoritma Logika Fuzzy pada sistem ini didasarkan pada variabel yang akan dikompensasi nantinya, yang berupa *setpoint* suhu dengan *disturbance* tekanan vakum. Dengan beberapa *input* yang masuk ke dalam kontroler, sangat memungkinkan untuk memakai Algoritma Logika Fuzzy. Selain tidak memerlukan perhitungan yang rumit untuk penentuan parameternya, Algoritma Logika Fuzzy juga memudahkan untuk membuat keputusan dari *input* yang sudah ada melalui aturan-aturan yang nantinya akan dibuat. Oleh karena itu, Algoritma Logika Fuzzy dipilih sebagai pengontrol suhu pada proses distilasi vakum bioetanol.

Pada skripsi ini akan dibuat suatu sistem pengontrolan suhu dengan struktur *Feedback-Feedforward* yang di aplikasikan pada proses distilasi vakum bioetanol dengan Algoritma Logika Fuzzy sebagai kontrolernya.

1.2 Rumusan Masalah

Berdasarkan uraian dalam latar belakang, didapatkan rumusan masalah sebagai berikut:

1. Bagaimana merancang dan membuat sistem pengontrolan suhu dengan struktur *Feedback-Feedforward* pada proses distilasi vakum?
2. Bagaimana merancang *hardware* dan *software* sistem pengontrolan suhu dengan struktur *Feedback-Feedforward* pada proses distilasi vakum berbasis Algoritma Logika Fuzzy?

1.3 Batasan Masalah

Karena luasnya objek pengkajian, maka perlu dilakukan pembatasan masalah agar pembahasan lebih terfokus pada rumusan masalah. Adapun batasan masalah pada skripsi ini antara lain:

1. *Vacuum Distiller* yang digunakan adalah *prototype* dengan desain sendiri.
2. *Feedback-Feedforward Control* yang diterapkan meliputi variabel *input* suhu dan tekanan vakum sebagai *disturbance*.
3. Pengontrolan suhu diatur dengan menggunakan elemen pemanas.
4. Bahan baku yang digunakan sebanyak 15 liter fermentasi tetes tebu.
5. Media pemanas di dalam jaket adalah minyak goreng sebanyak 15 liter.
6. Digunakan 4 buah elemen pemanas dengan tipe yang sama dengan daya total 1200W.
7. Sensor yang digunakan adalah sensor suhu *K Type Thermocouple* dan sensor tekanan MPX5100AP.
8. Suhu di dalam kondenser dikendalikan pada 62° celcius.
9. Kinerja *driver* dan eletronika tidak dibahas mendalam.
10. Mekanisme terjadinya Bioetanol tidak dibahas secara mendalam.

1.4 Tujuan

Tujuan dari skripsi ini adalah untuk merancang sistem pengontrolan suhu dengan struktur *Feedback-Feedforward* pada proses distilasi vakum berbasis Algoritma Logika *Fuzzy* sehingga nantinya diharapkan *output* yang dihasilkan sesuai dengan *setpoint* atau bisa menjaga suhu sama dengan *setpoint*.

1.5 Sistematika Pembahasan

Skripsi ini terdiri dari enam bab, dengan sistematika pembahasan sebagai berikut:

BAB I

Pendahuluan

Bab ini menguraikan tentang latar belakang, rumusan masalah, batasan masalah, tujuan, serta sistematika pembahasan.

BAB II

Tinjauan Pustaka

Tinjauan pustaka membahas teori-teori yang menunjang dalam perancangan dan pembuatan alat. Pustaka yang diambil adalah pustaka yang relevan dan sesuai serta mendukung penelitian, seperti buku-buku ilmu pengaturan dan lain-lain. Selain dari buku pustaka juga akan diambil dari jurnal, internet, dan sumber pengetahuan yang lain.

BAB III Metodologi Penelitian

Bab ini membahas tentang metode yang digunakan dalam pengerjaan dan perencanaan alat.

BAB IV Perancangan dan Pembuatan Alat

Bab ini membahas tentang perancangan alat yang meliputi spesifikasi, perencanaan sistem dan diagram balok, prinsip kerja, dan pembuatan alat serta perancangan *hardware* dan *software*. Setelah itu, bagaimana penerapannya dalam sistem secara keseluruhan.

BAB V Pengujian dan Analisis Sistem

Bab ini membahas tentang hasil pengujian sistem yang sudah dibuat, serta analisis hasil yang diperoleh.

BAB VI Kesimpulan dan Saran

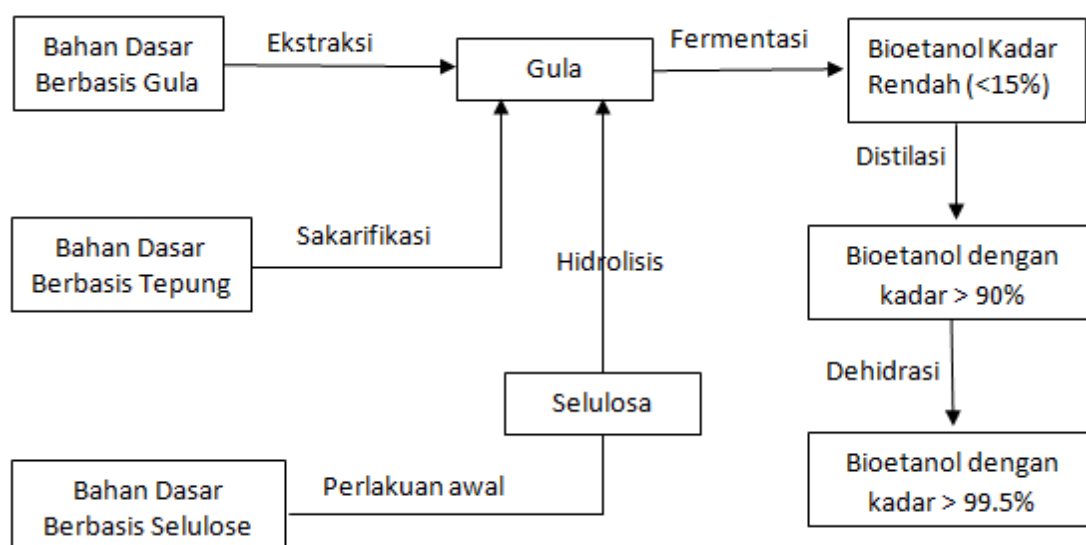
Dalam bab ini, semua hal yang sudah dikerjakan pada bab sebelumnya, dianalisis dan diambil kesimpulan. Serta rekomendasi dan saran untuk pengembangan alat selanjutnya.

BAB II

TINJAUAN PUSTAKA

2.1 Bioetanol

Bioetanol dapat diproduksi dari bahan-bahan nabati yang memiliki kandungan gula. Gambar 2.1 memperlihatkan 3 bahan dasar untuk memproduksi bioetanol. Bahan dasar dasar yang pertama adalah bahan dasar yang berbasis gula. Masuk dalam kategori ini adalah tebu dan *sweet sorghum*. Selain tebu itu sendiri, hasil sampingan dari proses produksi gula, yaitu tetes tebu (*molase*) dan *bagasse*. Bahan dasar yang kedua adalah berbasis tepung. Pada umumnya bahan makanan pokok manusia (jagung, kentang, singkong, dsb) masuk dalam kategori ini. Bahan dasar yang ketiga berbasis selulosa. Masuk dalam kategori ini adalah kayu, rumput dan sisa hasil panen. Dari ketiga bahan dasar tersebut, setelah terbentuknya gula akan menjalani proses yang sama untuk menjadi bioetanol, yaitu fermentasi dan distilasi.



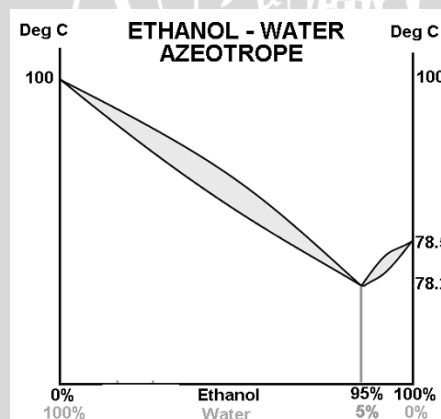
Gambar 2.1 Proses Dasar Pembuatan Bioetanol
Sumber: C. E. Wyman, dkk, 1996

2.2 Distilasi

Distilasi adalah suatu metode pemisahan campuran yang didasarkan pada perbedaan tingkat volatilitas (kemudahan suatu zat untuk menguap) pada suhu dan tekanan tertentu. Distilasi merupakan proses fisika dan tidak terjadi adanya reaksi kimia selama proses berlangsung.

Ada beberapa macam distilasi antara lain distilasi biasa, vakum, bertingkat, dan *azeotrope*. Distilasi biasa dilakukan dengan cara menaikkan suhu dan tekanan uapnya diatas larutan, sedangkan pada distilasi vakum penguapan cairan dilakukan pada tekanan rendah. Beda halnya dengan distilasi biasa dan vakum, kalau distilasi bertingkat banyak digunakan untuk memisahkan campuran zat cair dengan perbedaan titik didih yang tidak begitu besar dengan rentang tidak lebih dari 30°C. Pada proses ini ada proses yang disebut dengan proses *refluks* yang terjadi di kolom fraksinasi, yaitu kolom tempat kontak antara cairan dengan uap agar bertahan lebih lama sehingga komponen yang lebih rendah titik didihnya akan terus menguap ke pendingin atau kondensor. Distilasi bertingkat ini dilakukan pada tekanan tetap. Dan yang terakhir, yaitu pada distilasi *azeotrope* dilakukan penguapan tanpa perubahan komposisi.

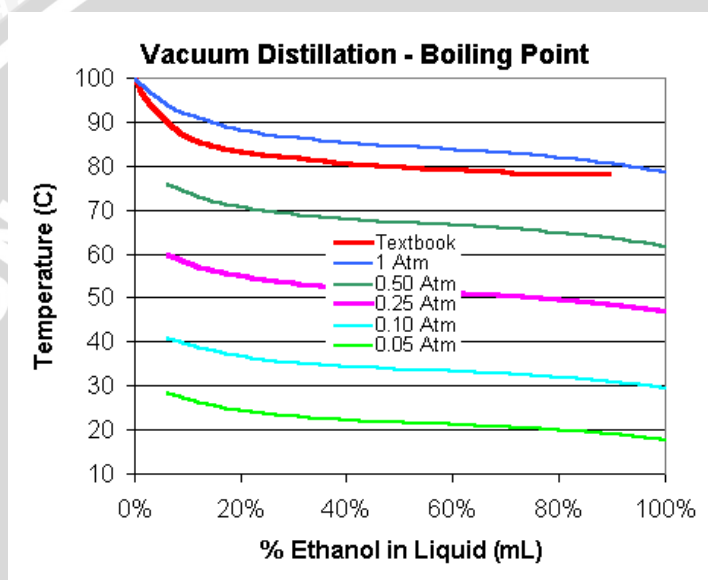
Secara teoritis proses distilasi (sebagaimana tercantum dalam Gambar 2.1) tidak akan dapat menghasilkan bioetanol dengan kadar diatas 95%. Hal ini dikarenakan fenomena terbentuknya *azeotrope* dari air dan etanol. Campuran ini memiliki titik didih yang lebih rendah dari titik uap air murni (100°C pada tekanan 1 atmosfer) dan titik didih etanol murni (78,5°C pada tekanan 1 atmosfer). Gambar 2.2 memperlihatkan fenomena *azeotrope* pada tekanan 1 atmosfer.



Gambar 2.2 Titik Didih Campuran Etanol-Air pada Tekanan 1 Atm
Sumber: Nixon dan Gaw, 2001

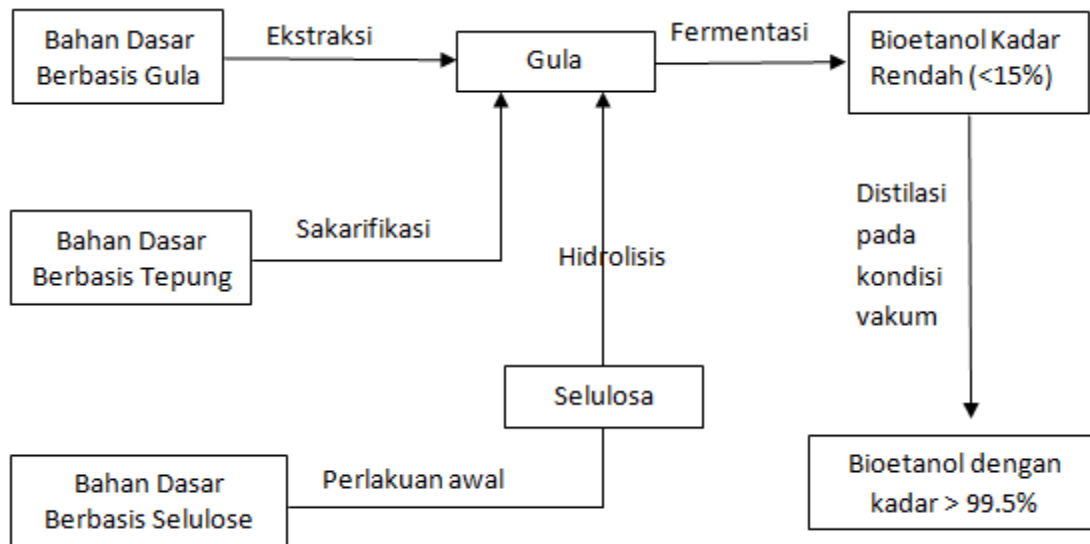
Untuk menghasilkan bioetanol dengan kadar sampai 100 %, pada umumnya dilakukan dengan proses dehidrasi menggunakan zat pengabsorpsi air, seperti *zeolit* dan garam. Proses ini akan memakan waktu 2 sampai 3 hari. Proses distilasi pada tekanan dibawah 1 atmosfer memberikan hasil konsentrasi bioetanol yang berbeda. Gambar 2.3 menggambarkan hal ini. Pada tekanan kurang dari 1 atmosfer, titik didih air murni dan alkohol akan turun dari kondisi normalnya. Dalam Gambar 2.3

diperlihatkan bahwa pada tekanan 0.1 atmosfer etanol murni (100%) akan didapatkan pada suhu sekitar 30°C (T. Ackland, 2012). Proses distilasi vakum memiliki kelemahan lain, yaitu titik didih air murni, alkohol murni dan campuran keduanya sangat dekat. Sehingga diperlukan sistem pengendali suhu yang bekerja dengan presisi. Hal ini menyebabkan distilasi vakum tidak digunakan untuk industri komersial. Di sisi lain, suhu dan tekanan adalah dua variabel yang saling mempengaruhi satu sama lain (H.S. Farkade, dkk, 2012). Sehingga diperlukan satu mekanisme kendali suhu dan tekanan terpadu, agar proses distilasi vakum dapat menghasilkan bioetanol dengan kadar mendekati 100 %.



Gambar 2.3 Distilasi Alkohol pada Tekanan dibawah 1 Atm
Sumber: T. Ackland, 2012

Penelitian ini akan fokus pada proses distilasi dari hasil hasil fermentasi, yaitu distilasi atas bioetanol dengan kadar rendah (<15%). Proses distilasi akan dilakukan pada kondisi yang mendekati vakum seperti dalam Gambar 2.4, sehingga proses dehidrasi alkohol tidak diperlukan lagi. Keuntungan dari sistem ini adalah efisiensi dari sisi waktu (menghemat waktu dehidrasi selama 2-3 hari) dan perbaikan kualitas etanol yang dihasilkan (prosentase bioetanol mendekati 100 %).



Gambar 2.4 Proses Pembuatan Bioetanol dengan Distilasi Vakum

2.3 Sistem Pengontrolan

Sistem pengontrolan berfungsi untuk mengontrol jalannya proses agar variabel proses yang sedang diukur dapat dikontrol dan diatur sesuai dengan nilai yang dikehendaki (*setpoint*). Gambar 2.5 memberikan kegiatan yang dilakukan dalam pengukuran, antara lain:



Gambar 2.5 Diagram Proses Pengukuran
Sumber: K. Ogata, 1985

Sistem kontrol mempunyai beberapa persyaratan umum, antara lain:

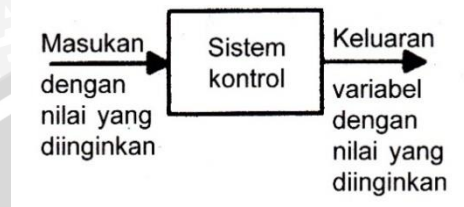
- Setiap sistem kontrol harus stabil.
- Sistem kontrol harus mempunyai kestabilan *relative* yang layak.
- Kecepatan respon harus cukup cepat dan menunjukkan peredaman yang layak.
- Suatu sistem kontrol juga harus mampu memperkecil kesalahan sampai nol atau sampai pada suatu harga yang dapat ditoleransi. (Khairul, 2014)

2.4 Loop Pengontrolan

Secara umum bentuk *loop* sistem pengontrolan dibagi menjadi dua macam, yaitu:

1. Sistem Pengontrolan *Loop* Terbuka (*Open Loop Control System*)

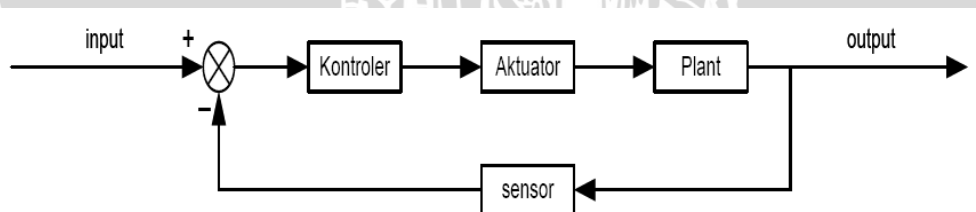
Sistem pengontrolan *loop* terbuka adalah sistem pengontrolan yang keluarannya tidak dapat mempengaruhi aksi dari pengontrolannya. Jadi, pada sistem ini keluaran dari kontrolernya tidak diukur atau diumpanbalikkan untuk membandingkan dengan masukannya (K. Ogata, 1985), seperti ditampilkan dalam Gambar 2.6.



Gambar 2.6 Diagram Balok Sistem *Loop* Terbuka
Sumber: W. Bolton, 2004

2. Sistem Pengontrolan *Loop* Tertutup (*Close Loop Control System*)

Sistem pengontrolan *loop* tertutup adalah sistem pengontrolan yang sinyal keluarannya mempunyai pengaruh langsung pada aksi pengontrolannya (K. Ogata, 1985). Sinyal kesalahan yang bekerja yaitu antara sinyal masukan dan sinyal umpan balik yang disajikan ke kontroler disajikan sedemikian rupa untuk mengurangi kesalahan dan membawa keluaran sistem ke nilai yang dikehendaki. Diagram balok sistem *loop* tertutup ditampilkan dalam Gambar 2.7.



Gambar 2.7 Diagram Balok *Close Loop*
Sumber: K. Ogata, 1985

Di dalam pengontrolan *loop* tertutup terdapat tiga macam metode pengontrolan, yaitu:

a. *Feed Back Control System*

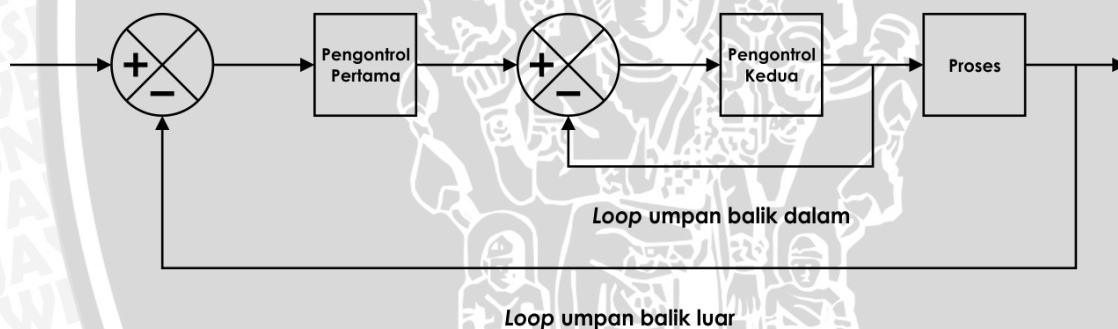
Feed back control system adalah sistem pengontrolan dimana besaran proses yang diatur dan diukur (PV) dibandingkan dengan nilai yang dikehendaki (SV) dan perbedaannya digunakan sebagai dasar untuk mengeliminir perbedaan yang ada.

b. Feed Forward Control System

Pada metode ini beban proses pengaturan diukur kemudian dibandingkan dengan beban normal dan bila ada perbedaan, maka perbedaan tersebut digunakan sebagai dasar untuk melakukan aksiantisipasi agar tidak terjadi penyimpangan pada *primary* proses variabel yang diatur.

c. Cascade Control System

Kontrol *cascade* adalah kontrol yang melibatkan penggunaan dua buah pengontrol dengan keluaran dari pengontrol pertama merupakan titik pengaturan bagi pengontrol kedua. *Loop* umpan balik untuk salah satu pengontrol berada di dalam *loop* umpan balik untuk pengontrol yang lain, seperti yang ditampilkan dalam Gambar 2.8. Sistem seperti ini dapat menghasilkan perbaikan karakteristik respon sistem terhadap gangguan yang muncul (W. Bolton, 2006: 290).



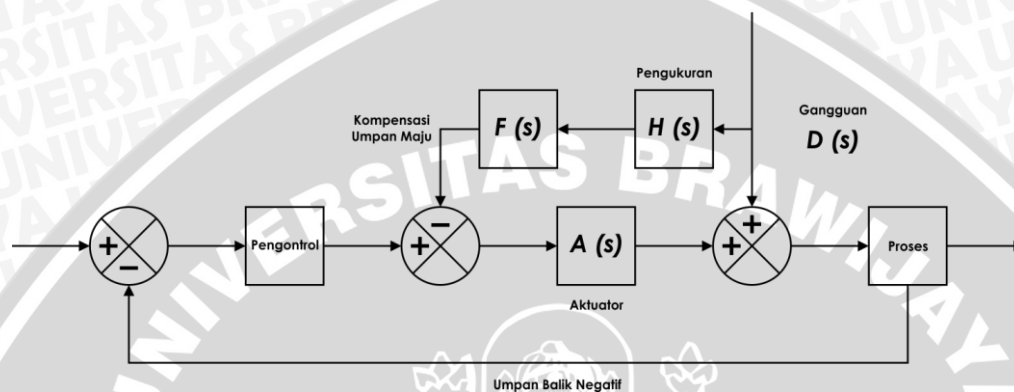
Gambar 2.8 Cascade Control System
Sumber: K. Ogata, 1985

Kontrol kaskade meliputi penggabungan dua kontroler menjadi satu yang difungsikan secara bersamaan dan dikontrol secara bersama pula. (Khairul, 2014)

2.5 Feedforward Control System

Pada sistem kontrol *loop* tertutup dasar, suatu sinyal gangguan harus berpropagasi melalui proses dan muncul sebagai sebuah masukan sinyal *error* untuk elemen pengontrol sebelum aksi pengontrolan diambil untuk mengoreksinya (W. Bolton, 2004). Ini berarti terdapat suatu *deviasi* atau penyimpangan dari titik pengaturan saat propagasi melalui sistem terjadi dan aksi kontrol diambil. Untuk

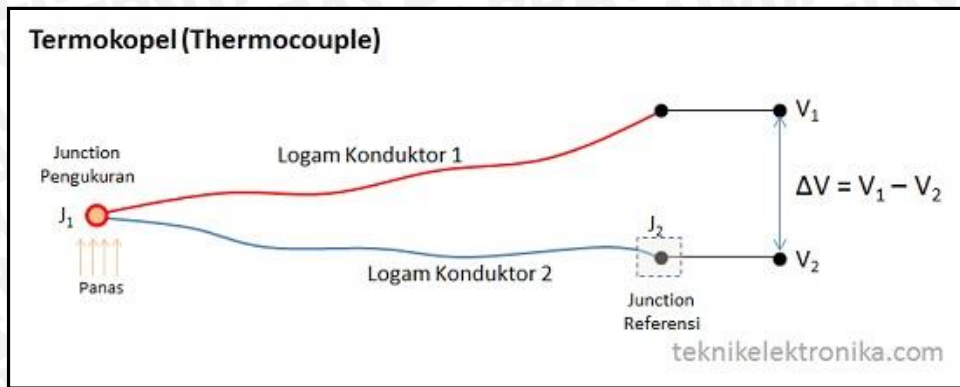
beberapa sistem kontrol, hal semacam ini tidak bisa diterima. Untuk mengatasinya, maka digunakan kontrol umpan maju (*Feedforward Control*). Dengan bentuk kontrol semacam ini, sinyal gangguan diukur dan sebuah sinyal ditambahkan ke keluaran pengontrol untuk memacu moda kontrol. Jadi, aksi korektif dimulai tanpa menunggu efek dari gangguan muncul pada masukan sinyal *error* ke elemen pengontrol. Gambar 2.9 menunjukkan sistem kontrol umpan maju (*Feedforward Control System*) tersebut.



Gambar 2.9 Sistem Kontrol Umpan Maju (*Feedforward Control System*)
Sumber: W. Bolton, 2004

2.6 Sensor Suhu

Untuk mengetahui perubahan panas yang terjadi di dalam tabung evaporator, maka digunakanlah sensor suhu. Sensor suhu yang digunakan disini adalah *K Type Thermocouple*, yang merupakan salah satu jenis sensor suhu yang digunakan untuk mendeteksi suhu melalui dua jenis logam konduktor berbeda yang digabung pada ujungnya sehingga menimbulkan efek dimana sebuah logam konduktor yang diberi perbedaan panas secara gradient akan menghasilkan tegangan listrik, dengan satu jenis logam konduktor yang terdapat pada *thermocouple* akan berfungsi sebagai referensi dengan suhu tetap, sedangkan yang satunya lagi sebagai logam konduktor yang mendeteksi suhu panas. Besar tegangan listrik yang terbentuk tergantung dari jenis material konduktor yang digunakan, serta besar perbedaan suhu antara dua konduktor tersebut. Sinyal *output* dari *thermocouple* adalah berupa tegangan listrik yang bernilai milivolt. Penjelasan mengenai prinsip kerja *thermocouple* dapat dilihat dalam Gambar 2.10.



Gambar 2.10 Prinsip Kerja *Thermocouple*
Sumber: www.teknikelektronika.com

Thermocouple yang dipakai pada skripsi ini adalah *K Type Thermocouple* dimana *thermocouple* tersebut mempunyai bahan logam konduktor positif : Nickel – Chromium dan bahan logam konduktor negatif : Nickel – Alumunium. *K Type Thermocouple* mempunyai rentang suhu antara -200°C sampai 1250°C yang membuat sensor ini sangat berguna untuk sistem dengan range suhu besar. Gambar 2.11 merupakan tampilan luar dari *K Type Thermocouple*.



Gambar 2.11 *K Type Thermocouple*
Sumber: <http://www.ebay.ca/itm/All-purpose-Temperature-Controller-STC-1000-Whit-sensor-/320628773067>

2.7 *Thermocouple Amplifier* MAX6675

Thermocouple Amplifier MAX6675 ini suatu modul rangkaian yang berfungsi sebagai penguat tegangan serta sebagai ADC dari masukan sensor suhu (*K Type Thermocouple*) ke keluaran modul arduino mega 2560. Penampang dari *Thermocouple Amplifier* MAX6675 kecil tetapi memiliki 8 pin yang diantaranya yaitu ground, input sensor (+), input sensor (-), VCC, SCK, CS, SO, dan NC. Modul ini memungkinkan untuk membaca suhu sampai 1024°C serta dengan resolusi tiap perubahan suhu bisa mencapai $0,25^{\circ}\text{C}$ yang sangat memungkinkan dan sangat

berguna sebagai amplifier untuk sensor suhu *K Type Thermocouple*. Tampilan modul *Thermocouple Amplifier* MAX6675 ditunjukkan dalam Gambar 2.12.



Gambar 2.12 *Thermocouple Amplifier* MAX6675

Sumber: www.indo-ware.com/produk-2951-max6675-thermocouple-modul.html

2.8 Sensor Tekanan

Untuk mengetahui besar tekanan udara vakum yang ada pada proses distilasi vakum, maka diperlukan sensor tekanan vakum, yaitu sensor tekanan MPX5100AP. Sensor tekanan MPX5100AP merupakan sensor dengan tipe absolut (*absolute type*) yang berarti bahwa pengukuran pada tipe sensor ini dilakukan dengan membandingkan tekanan yang diukur dengan tekanan vakum (*vacuum pressure*). Sensor ini dapat dipergunakan untuk mengukur tekanan dengan range antara 0 – 100 kPa. Untuk dapat bekerja, sensor ini memerlukan *power supply* sebesar 5 VDC, serta *output* yang berupa tegangan VDC yang nantinya masuk ke dalam modul Arduino Mega 2560 untuk membuat algoritma pengendaliannya. Gambar 2.13 adalah tampilan luar dari sensor tekanan MPX5100AP.



Gambar 2.13 Sensor Tekanan MPX5100AP

Sumber: www.octopart.com

2.9 Elemen Pemanas (*Heater*)

Elemen pemanas merupakan piranti yang mengubah energi listrik menjadi energi panas melalui proses *Joule Heating*. Prinsip kerja elemen panas adalah arus listrik yang mengalir pada elemen menjumpai resistansinya, sehingga menghasilkan panas pada elemen.

Panas yang dihasilkan oleh elemen pemanas listrik ini bersumber dari kawat ataupun pita bertahanan listrik tinggi (*Resistance Wire*) biasanya bahan yang digunakan adalah niklin yang dialiri arus listrik pada kedua ujungnya dan dilapisi oleh isolator listrik yang mampu meneruskan panas dengan baik hingga aman jika digunakan.

Ada 2 macam jenis utama pada elemen pemanas listrik ini yaitu:

- Elemen Pemanas Listrik bentuk dasar seperti yang ditunjukkan dalam Gambar 2.14, yaitu elemen pemanas dimana *Resistance Wire* hanya dilapisi oleh isolator listrik, macam-macam elemen pemanas bentuk ini adalah: Ceramic Heater, Silica Dan Quartz Heater, Bank Channel heater, Black Body Ceramic Heater.



Gambar 2.14 Elemen Pemanas Listrik Bentuk Dasar
Sumber: www.custom-electric.com

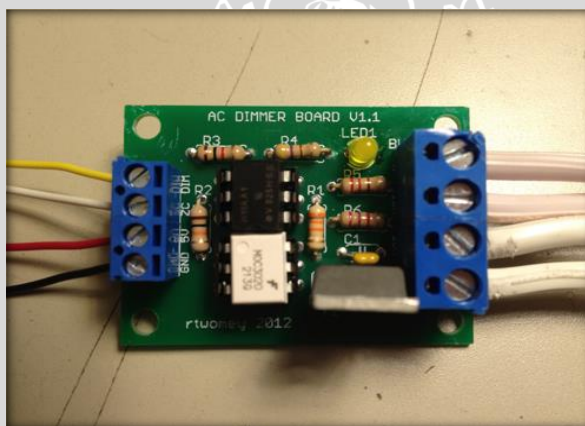
- Elemen pemanas listrik bentuk lanjut yang ditunjukkan dalam Gambar 2.15, merupakan elemen pemanas dari bentuk dasar yang dilapisi oleh pipa atau lembaran plat logam untuk maksud sebagai penyesuain terhadap penggunaan dari elemen pemanas tersebut. Bahan logam yang biasa digunakan adalah: *mild stell*, *stainless stell*, tembaga, dan kuningan. (Akhmad Salmi F., 2014)



Gambar 2.15 Elemen Pemanas Listrik Bentuk Lanjut
Sumber: WWW.spares4appliances.co.uk

2.10 Rangkaian Dimmer AC

Diperlukan sebuah rangkaian yang bisa digunakan untuk melakukan pengendalian AC, rangkaian tersebut yaitu rangkaian *dimmer* AC yang di dalamnya terdiri dari rangkaian TRIAC dan rangkaian *Zero Cross Detector*. Tampilan Modul *dimmer* AC ditunjukkan dalam Gambar 2.16.



Gambar 2.16 AC Dimmer Modul Lite v1.1
Sumber: T. Robert, 2013

2.10.1 Rangkaian TRIAC

TRIAC atau *Triode for Alternating Current* (Trioda untuk arus bolak-balik) adalah komponen yang tersusun atas dua buah *thyristor* antiparalel dengan terminal *gate* yang sama. Nama resmi untuk TRIAC adalah *Bidirectional Triode Thyristor*. Ini menunjukkan saklar dwiarah yang dapat mengalirkan arus listrik ke kedua arah ketika dipicu (dihidupkan). Konfigurasi tersebut membuat TRIAC merupakan komponen yang digunakan untuk mengatur tegangan AC sehingga pemicuan dari gate TRIAC juga dapat dilakukan dengan pulsa negatif atau pulsa positif (sesuai tegangan AC) (Primatama, 2013).

TRIAC dalam penggunaannya membutuhkan DIAC sebagai *driver*. Dibutuhkan juga komponen yang mampu berfungsi sebagai isolator antara tegangan 220V dan 5V untuk melindungi mikrokontroler dari tegangan balik. Sehingga komponen yang sesuai dengan spesifikasi di atas adalah *optocoupler*. *Optocoupler* yang komponen utamanya berupa DIAC adalah MOC3022.

2.10.2 Rangkaian Zero Cross Detector

Zero crossing detector adalah rangkaian yang digunakan untuk mendeteksi gelombang sinus AC 220 volt saat melewati titik tegangan nol. Seberangan titik nol yang dideteksi adalah peralihan dari positif menuju negatif dan peralihan dari negatif menuju positif. Seberangan-seberangan titik nol ini merupakan acuan yang digunakan sebagai awal pemberian nilai waktu tunda untuk pemicuan TRIAC (Primatama, 2013).

Rangkaian *zero cross detector* membutuhkan rangkaian yang bisa mendeteksi persimpangan antara gelombang tegangan bolak-balik dengan titik nol agar pengaturan fasa dapat dilakukan pada titik nol gelombang tegangan bolak-balik. Metode ini berfungsi untuk menentukan frekuensi suatu gelombang dengan cara mendeteksi banyaknya zero point pada suatu rentang waktu. *Zero crossing detector* berfungsi untuk mendeteksi perpotongan gelombang sinus pada tegangan AC dengan *zero point* tegangan AC tersebut, sehingga dapat memberikan sinyal acuan saat dimulainya pemicuan sinyal PWM. Dengan rangkaian *zero crossing detector*, kita bisa mendeteksi *zero point* sekaligus mengubah suatu sinyal sinusoida menjadi sinyal kotak. Sinyal keluaran rangkaian *zero cross detector* ini akan dimasukkan ke mikrokontroler, oleh karena itu dibutuhkan juga komponen yang mampu memisahkan tegangan 5V dan 220V. Rangkaian ini menggunakan komponen *optocoupler* 4N25. (Akhmad Salmi F., 2014)

2.11 Algoritma Logika Fuzzy

Fuzzy secara harfiah berarti samar, sedangkan kebalikannya dalam hal ini adalah *Crisp* yang secara harfiah berarti tegas. Dalam kehidupan sehari-hari nilai samar lebih akrab daripada nilai tegas. Temperatur/suhu tertentu biasa dinyatakan

sebagai panas, agak panas, atau sangat dingin daripada dinyatakan dalam nilai terukur tertentu.

Tahun 1965 L.A. Zadeh memodifikasi teori himpunan yang disebut himpunan kabur (*Fuzzy Set*). Himpunan *fuzzy* didasarkan pada gagasan untuk memperluas jangkauan fungsi karakteristik sehingga fungsi tersebut akan mencakup bilangan real pada interval $[0, 1]$. Nilai keanggotaannya menunjukkan bahwa suatu nilai dalam semesta pembicaraan tidak hanya berada pada 0 atau 1, namun juga nilai yang terletak diantaranya. Dengan kata lain nilai kebenaran suatu hal tidak hanya bernilai benar atau salah. Nilai 0 menunjukkan salah, nilai 1 menunjukkan benar dan masih ada nilai-nilai yang terletak diantaranya.

Sejak tahun 1985 pengaturan berbasis logika *fuzzy* mengalami perkembangan pesat, terutama dalam hubungannya dengan penyelesaian masalah kendali yang bersifat tak linier, sulit dimodelkan, berubah karakteristiknya terhadap waktu (*time varying*) dan kompleks.

2.11.1 Struktur Dasar Logika *Fuzzy*

Dalam sistem pengaturan dengan logika *fuzzy* dilibatkan suatu blok pengontrol yang menerima satu atau lebih masukan dan mengumpankan satu atau lebih keluaran ke plant atau blok lain. Komponen utama penyusun Algoritma Logika *Fuzzy* adalah unit fuzzifikasi, *fuzzy inference*, *rule base*, dan unit defuzzifikasi.

Rule Base terdiri dari dua jenis (Yan,1994):

- Basis data

Mendefinisikan parameter *fuzzy* sebagai bagian dari himpunan *fuzzy* dengan menentukan batas-batas fungsi keanggotaan pada semesta pembicaraan untuk tiap-tiap variabel.

- Basis aturan

Memetakan nilai masukan *fuzzy* menjadi nilai keluaran *fuzzy*.

2.11.2 Fungsi Keanggotaan

Fungsi keanggotaan menotasikan nilai kebenaran anggota-anggota himpunan *fuzzy*. Interval nilai yang digunakan untuk menentukan fungsi keanggotaan, yaitu nol dan satu. Tiap fungsi keanggotaan memetakan elemen himpunan *crisp* ke semesta himpunan *fuzzy*.

Suatu himpunan *fuzzy* A dalam semesta pembicaraan U dinyatakan dengan fungsi keanggotaan, μ_A yang harganya berada dalam interval [0,1] (Kuswadi, 2000:27). Secara matematika hal ini dinyatakan dalam Persamaan (2-1) berikut:

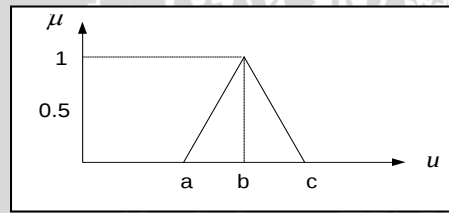
$$\mu_A : U \rightarrow [0,1] \quad (2-1)$$

- Fungsi keanggotaan bentuk Triangular

Definisi fungsi triangular ditunjukkan dalam Persamaan (2-2) berikut:

$$T(u; a, b, c) = \begin{cases} 0 & u < a \\ \frac{u-a}{b-a} & a \leq u \leq b \\ \frac{c-u}{c-b} & b \leq u \leq c \\ 0 & u > c \end{cases} \quad (2-2)$$

Fungsi keanggotaan bentuk triangular ditunjukkan dalam Gambar 2.17.



Gambar 2.17 Fungsi Keanggotaan Bentuk Triangular
Sumber: Yan, 1994

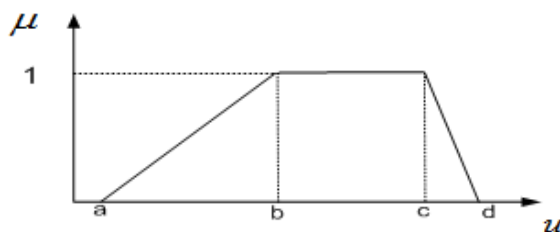
Fungsi keanggotaan bentuk triangular ini digunakan bila diinginkan himpunan *fuzzy* mempunyai nilai proporsional terhadap nol maupun satu.

- Fungsi keanggotaan bentuk Trapezium

Definisi fungsi trapesium ditunjukkan dalam Persamaan (2-3) berikut:

$$T(u; a, b, c, d) = \begin{cases} 0 & u < a \\ \frac{u-a}{b-a} & a \leq u \leq b \\ 1 & b \leq u \leq c \\ \frac{d-u}{d-c} & c \leq u \leq d \\ 0 & d \leq u \end{cases} \quad (2-3)$$

Fungsi keanggotaan bentuk trapesium ditunjukkan dalam Gambar 2.18.



Gambar 2.18 Fungsi Keanggotaan Bentuk Trapesium
Sumber: Yan, 1994

2.11.3 Algoritma Logika Fuzzy

Algoritma Logika Fuzzy adalah sistem berbasis aturan (*rule based system*) yang didalamnya terdapat himpunan aturan fuzzy yang mempresentasikan mekanisme pengambilan keputusan. Aturan yang dibuat digunakan untuk memetakan variabel *input* ke variabel *output* dengan pernyataan *If – Then*.

Algoritma Logika Fuzzy ini akan menggunakan data tertentu (*crisp*) dari sejumlah sensor kemudian mengubahnya menjadi bentuk linguistik atau fungsi keanggotaan melalui proses fuzzifikasi. Lalu dengan aturan fuzzy, *inference engine* yang akan menentukan hasil keluaran fuzzy. Setelah itu hasil ini akan diubah kembali menjadi bentuk numerik melalui proses defuzzifikasi.

2.11.3.1 Fuzzifikasi

Proses fuzzifikasi merupakan proses untuk mengubah variabel non fuzzy (variabel numerik) menjadi variabel fuzzy (variabel linguistik). Nilai masukan-masukan yang masih dalam bentuk variabel numerik yang telah dikuantisasi sebelum diolah oleh pengontrol logika fuzzy harus diubah terlebih dahulu ke dalam variabel fuzzy. Melalui fungsi keanggotaan yang telah disusun, maka dari nilai-nilai masukan tersebut menjadi informasi fuzzy yang berguna nantinya untuk proses pengolahan secara fuzzy pula. Proses ini disebut fuzzifikasi (Yan,1994). Proses fuzzifikasi diekspresikan sebagai berikut:

$X = \text{fuzzifier}(x_0)$

dengan:

x_0 = nilai *crisp* variabel masukan

x = himpunan *fuzzy* variabel yang terdefinisi

fuzzifier = operator fuzzifikasi yang memetakan himpunan *crisp* ke himpunan *fuzzy*

Pedoman memilih fungsi keanggotaan untuk proses fuzzifikasi, menurut Jun Yan, menggunakan:

1. Himpunan *fuzzy* dengan distribusi simetris.
2. Gunakan himpunan *fuzzy* dengan jumlah ganjil, berkaitan erat dengan jumlah kaidah (*rules*).
3. Mengatur himpunan *fuzzy* agar saling menumpuk.
4. Menggunakan fungsi keanggotaan bentuk segitiga atau trapesium.

2.11.3.2 Kaidah Aturan Fuzzy (Fuzzy Rule)

Fuzzy rule adalah bagian yang menggambarkan dinamika suatu sistem terhadap masukan yang dikarakteristikan oleh sekumpulan variabel-variabel linguistik dan berbasis pengetahuan seorang operator ahli. Pernyataan tersebut umumnya dinyatakan oleh suatu pernyataan bersyarat.

Dalam pengontrol berbasis *fuzzy*, aturan pengontrolan *fuzzy* berbentuk aturan “IF – THEN”. Untuk sebuah sistem *Multi Input Single Output* (MISO) basis aturan pengendalian *fuzzy* berbentuk seperti berikut ini:

Rule 1 IF X is A_1 AND Y is B_1 THEN Z is C_1

Rule 2 IF X is A_2 AND Y is B_2 THEN Z is C_2

⋮

Rule n IF X is A_n AND Y is B_n THEN Z is C_n

Dengan X, Y, Z merupakan variabel linguistik, dimana X dan Y merupakan variabel masukan, dan Z merupakan variabel keluaran sistem. $A_n, B_n,$ dan C_n merupakan nilai linguistik dari $X, Y,$ dan Z (Lee, 1990).

2.11.3.3 Metode Inferensi MAX-MIN

Metode inferensi merupakan proses untuk mendapatkan keluaran dari suatu kondisi masukan dengan mengikuti aturan-aturan yang telah ditetapkan. Keputusan yang didapatkan pada proses ini masih dalam bentuk *fuzzy* yaitu derajat keanggotaan keluaran.

Pada metode Max-Min aturan operasi minimum Mamdani digunakan untuk implikasi *fuzzy* yang ditunjukkan dalam Persamaan (2-4) dan Persamaan (2-5). Persamaan aturan minimum adalah:

$$\mu_{C'} = \bigcup_1^n \alpha_i \wedge \mu_{Ci} \quad (2-4)$$

$$\text{dengan } \alpha_i = \mu_{Ai}(x_0) \wedge \mu_{Bi}(y_0) \quad (2-5)$$

Sebagai contoh, terdapat dua baris kaidah atur *fuzzy*, yaitu:

R₁ : Jika x adalah A₁ dan y adalah B₁ maka z adalah C₁

R₂ : Jika x adalah A₂ dan y adalah B₂ maka z adalah C₂

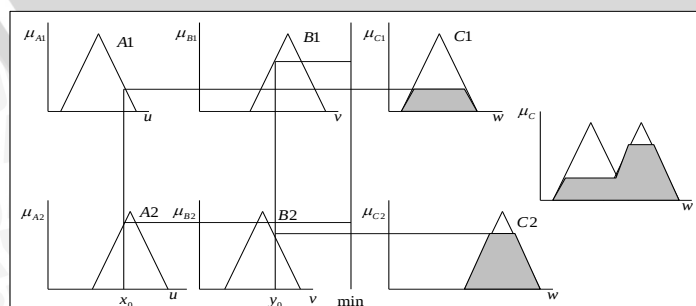
Pada metode penalaran MAX-MIN fungsi keanggotaan konsekuensi dinyatakan dalam Persamaan (2-6), Persamaan (2-7), dan Persamaan (2-8) berikut:

$$\mu_{C'}(w) = \mu_{C'1} \vee \mu_{C'2} = [\alpha_1 \wedge \mu_{C1}(w)] \vee [\alpha_2 \wedge \mu_{C2}(w)] \quad (2-6)$$

$$\text{dimana } \alpha_1 = \mu_{A1}(x_0) \wedge \mu_{B1}(y_0) \quad (2-7)$$

$$\alpha_2 = \mu_{A2}(x_0) \wedge \mu_{B2}(y_0) \quad (2-8)$$

Lebih jelas metode ini dideskripsikan dalam Gambar 2.19.



Gambar 2.19 Inferensi Fuzzy dengan Metode MAX-MIN
Sumber: Yan, 1994

2.11.3.4 Metode Defuzzifikasi *Weighted Average* (WA)

Defuzzifikasi adalah proses untuk mendapatkan nilai numerik dari data *fuzzy* yang dihasilkan dari proses inferensi (Yan, 1994). Proses defuzzifikasi dinyatakan dalam Persamaan (2-9) sebagai berikut:

$$y_0 = \text{defuzzifier}(y) \quad (2-9)$$

dengan:

y : aksi kontrol *fuzzy*

y_0 : aksi kontrol *crisp*

defuzzifier : operator defuzzifikasi

Metode *Weighted Average* (WA) ini didefinisikan dalam Persamaan (2-10) sebagai berikut:

$$U = \frac{\sum_{i=1}^n w_i u_i}{\sum_{i=1}^n w_i} \quad (2-10)$$

dengan:

U = Keluaran

w_i = Bobot nilai benar w_i

u_i = Nilai linguistik pada fungsi keanggotaan keluaran

n = Banyak derajat keanggotaan

2.12 Arduino Mega 2560

Arduino Mega 2560 adalah *board* mikrokontroler berbasis ATmega2560. Arduino Mega memiliki 54 pin *Input/Output* digital (dimana 15 pin diantaranya dapat digunakan sebagai output PWM), 16 input analog, 4 UARTs (port serial perangkat keras), 16 MHz osilator Kristal, koneksi USB, *power jack*, ICSP *header*, dan tombol reset. Untuk mendukung mikrokontroler agar dapat digunakan, cukup hanya menghubungkan Arduino Mega 2560 ke komputer dengan menggunakan kabel USB atau dengan memberi daya dengan adaptor AC-to-DC atau baterai agar dapat bekerja. Arduino Mega 2560 juga cocok menggunakan sebagian besar *shield* yang didesain untuk Arduino Duemilanove.

Arduino Mega 2560 berbeda dengan *board* Arduino sebelumnya dimana *board* sebelumnya menggunakan FTDI USB-to-serial *driver chip*. Pada Arduino Mega 2560 terdapat ATmega16U2 (sebagai ganti dari ATmega8U2 pada *board*

sebelumnya) yang deprogram sebagai USB-to-serial *converter*. Arduino Mega2560 juga memiliki RESET sirkuit yang lebih kuat dibandingkan *board* sebelumnya. Fitur baru yang terdapat pada Arduino Mega 2560 yaitu penambahan pin SDA dan SCL yang berdekatan dengan pin AREF dan dua pin baru lainnya diletakkan dekat pin RESET, pin pertama yaitu IOREF yang memperbolehkan shield untuk beradaptasi dengan tegangan yang disediakan oleh *board*. Kedepannya, *shield* akan bekerja sangat baik dengan *board* yang menggunakan AVR yang dapat beroperasi dengan 5V dan dengan 3,3V (seperti pada Arduino Duemilanove). Pin kedua adalah pin yang tidak terhubung yang disediakan untuk tujuan yang akan datang. Gambar 2.20 merupakan tampak depan dari Arduino Mega 2560.



Gambar 2.20 Board Arduino Mega 2560
Sumber: <https://store.arduino.cc/product/A000067>



BAB III

METODOLOGI PENELITIAN

Metodologi penelitian disini difungsikan demi tercapainya tujuan skripsi yang terdapat di bab pendahuluan. Berikut adalah langkah-langkah yang akan digunakan dalam penelitian:

3.1 Perancangan Alat

Perancangan alat dilakukan sebagai langkah awal sebelum terbentuknya suatu sistem beserta rangkaian elektrik (*hardware*) serta pemrogramannya (*software*). Hal ini dimaksudkan agar pengontrolan suhu pada proses distilasi vakum bioetanol dapat berjalan sesuai deskripsi awal yang telah direncanakan.

Perancangan alat yang dilakukan meliputi:

1. Penentuan spesifikasi alat, yang meliputi:
 - Penentuan deskripsi kerja alat keseluruhan.
 - Algoritma Logika *Fuzzy*, program serta rangkaian elektronik dan *software* pendukung.
2. Penentuan spesifikasi kontroler, yang meliputi:
 - *Error steady state* (e_{ss}).
 - *Maximum overshoot* (M_p).
 - *Settling time* (t_s).
3. Pemilihan rangkaian elektrik, meliputi:
 - Rangkaian *power supply*.
 - Modul *Thermocouple Amplifier* MAX6675.
 - Modul *Filter Sensor Tekanan*.
 - Rangkaian *dimmer* AC.
 - Rangkaian *Liquid Crystal Display* (LCD).
4. Perancangan fungsi keanggotaan dan *rule base*, serta metode *defuzzifikasi* untuk Algoritma Logika *Fuzzy* dan perancangan program Arduino Mega 2560 pada *software* Arduino ERW 1.6.5.

3.2 Realisasi Pembuatan Alat

Untuk merealisasikan alat yang telah dirancang sebelumnya, langkah-langkah yang dilakukan adalah sebagai berikut.

1. Pembuatan mekanik *vacuum distiller* secara keseluruhan yang meliputi pemilihan elemen pemanas, sensor suhu dan tekanan, motor pengaduk, pompa, dan komponen lainnya.
2. Perancangan spesifikasi kontroler yang meliputi besar *error steady state* (e_{ss}), *maximum overshoot* (M_p), dan lama *settling time* (t_s).
3. Pemilihan rangkaian elektrik meliputi modul *Thermocouple Amplifier MAX6675*, modul *filter* sensor tekanan, rangkaian *dimmer AC*, *power supply* dan LCD.
4. Pembuatan dan penyelesaian fungsi keanggotaan dan *rule base*, serta metode *defuzzifikasi* untuk Algoritma Logika Fuzzy dan program Arduino Mega 2560 pada *software Arduino ERW 1.6.5*.

3.3 Pengujian Alat

Setelah semua komponen pada alat sudah terhubung sesuai dengan diagram balok sistem yang telah dirancang dan perangkat lunak (*software*) untuk mendukung sistem yang telah dibuat, maka diadakan pengujian dan analisis alat.

Pengujian dan analisis yang dilakukan adalah sebagai berikut:

1. Pengujian Sensor *K Type Thermocouple*
2. Pengujian *Thermocouple Amplifier MAX6675*
3. Pengujian Sensor MPX5100AP
4. Pengujian LCD
5. Pengujian Sudut Penyalaan TRIAC
6. Pengujian Rangkaian *Dimmer AC*
7. Pengujian Sistem Keseluruhan

3.4 Pengambilan Kesimpulan

Kesimpulan diambil berdasarkan data yang didapat dari hasil pengujian sistem secara keseluruhan. Apabila hasil yang didapatkan sesuai dengan apa yang diharapkan sebelumnya, maka sistem kontrol tersebut sudah berhasil memenuhi tujuan (spesifikasi) awal dan selanjutnya dapat dimasukkan ke dalam saran guna dikembangkan untuk disempurnakan di penelitian selanjutnya.

BAB IV

PERANCANGAN DAN PEMBUATAN ALAT

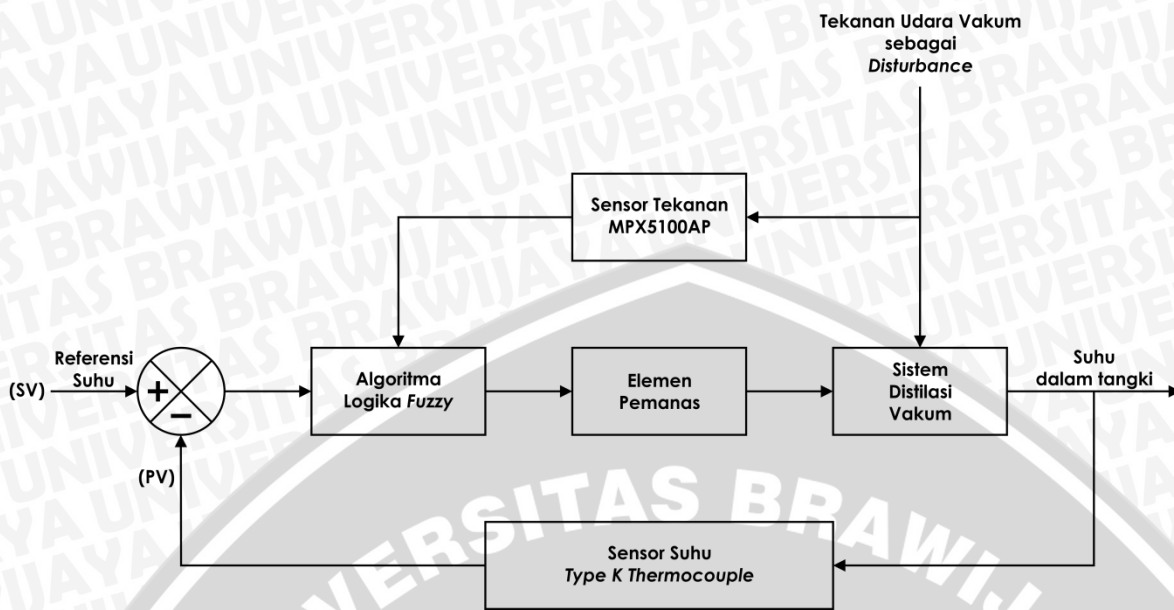
Dalam bab ini akan dibahas mengenai perancangan dan pemilihan alat untuk sistem pengontrolan suhu pada proses distilasi vakum bioetanol. Pada dasarnya perancangan alat meliputi pemilihan perangkat keras (*hardware*) dan perancangan perangkat lunak (*software*).

4.1 Perancangan Sistem

Perancangan dan pemilihan alat yang dilakukan secara bertahap untuk memudahkan analisis sistem. Beberapa aspek yang perlu dijelaskan dalam bab ini meliputi:

1. Spesifikasi alat.
2. Spesifikasi kontroler.
3. Prinsip kerja alat.
4. Pemilihan perangkat keras (mekanik *Vacuum Distiller*, modul *Thermocouple Amplifier MAX6675*, rangkaian *filter* sensor tekanan, elemen pemanas, rangkaian *dimmer AC*, *power supply*).
5. Perancangan perangkat lunak (perancangan Algoritma Logika *Fuzzy* dan *flowchart* sistem keseluruhan pada *software Arduino ERW 1.6.5*).

4.2 Diagram Balok Sistem



Gambar 4.1 Diagram Balok Sistem (Perancangan)

Keterangan dari diagram balok sistem dalam Gambar 4.1 adalah sebagai berikut:

- *Setpoint* sistem adalah suhu 62°C.
- Pusat pengontrolan sistem adalah Arduino Mega 2560 yang memberikan sinyal PWM pada rangkaian *dimmer* AC.
- Rangkaian *dimmer* AC berfungsi sebagai pengontrol sudut-sudut penyalan pada TRIAC dan juga PWM.
- Sensor suhu *K Type Thermocouple* berfungsi sebagai *feedback* (umpan balik) sistem yang dihubungkan dengan modul *Thermocouple Amplifier* MAX6675 agar menghasilkan tegangan yang sesuai dengan rentang tegangan masukan pusat pengendali (kontroler).
- Sensor tekanan berfungsi sebagai pendeteksi tekanan vakum yang ada di dalam *Vacuum Disiller*. Jenis sensor tekanan yang digunakan adalah MPX5100AP yang dicatu tegangan DC sebesar 5 V.
- Kontroler yang digunakan adalah Algoritma Logika Fuzzy dengan menggunakan Arduino Mega 2560 untuk mengolah Fuzzifikasi, *Rule-Base*, Inferensi, dan Defuzzifikasi yang digunakan sebagai sinyal kontrol berupa PWM.

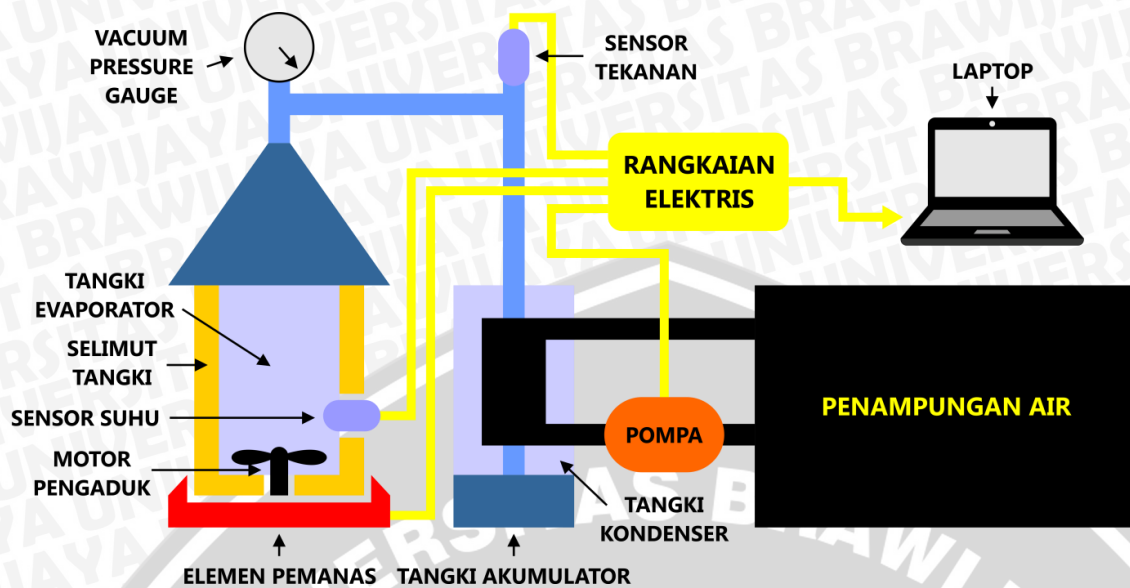
- Gangguan (*disturbance*) berupa tekanan udara vakum yang berada di dalam sistem.

4.3 Spesifikasi Alat

Adapun spesifikasi alat yang akan direalisasikan adalah sebagai berikut:

- Tangki larutan (Evaporator), merupakan tempat terjadinya proses penguapan (evaporasi) larutan (bioetanol kadar rendah).
- Tangki kondenser, merupakan tempat terjadinya proses kondensasi, yang disambungkan dengan hasil penguapan di tangki larutan.
- Tangki akumulator, merupakan tempat untuk menampung cairan hasil dari proses kondensasi.
- Elemen pemanas dengan daya total sebesar 1200 W, merupakan piranti yang digunakan untuk memanaskan minyak yang melapisi bagian luar (jaket) dari tangki evaporator.
- Piranti vakum, merupakan alat yang digunakan untuk menurunkan tekanan proses dibawah tekanan ruang 1 atmosfer pada tangki evaporator, kondenser, dan tangki akumulator.
- Perangkat kontrol yang digunakan adalah Arduino Mega 2560.
- Sensor yang digunakan adalah sensor suhu dan sensor tekanan.
- Sensor suhu yang digunakan adalah *K Type Thermocouple*, yang dipasang di dalam tangki sehingga bisa bersentuhan langsung dengan larutan bahan baku.
- Sensor tekanan yang digunakan yaitu MPX5100AP, yang dipasang diatas tangki kondenser sehingga bisa melihat perubahan tekanan vakum yang terjadi.
- Rangkaian *dimmer AC* yang digunakan sebagai pengontrol PWM yang masuk ke dalam elemen pemanas.
- Modul *Thermocouple Amplifier MAX6675* digunakan sebagai penguat tegangan serta sebagai ADC dari *output* sensor suhu (*K Type Thermocouple*) ke *input* modul Arduino Mega 2560.
- Kontroler yang digunakan ialah Algoritma Logika Fuzzy.

Skema perancangan sistem dapat dilihat dalam Gambar 4.2.



Gambar 4.2 Skema Sistem Keseluruhan (Perancangan)

4.4 Spesifikasi Kontrol

Penentuan spesifikasi kontrol disini sangat penting karena spesifikasi inilah yang nanti akan dijadikan dasar untuk melihat apakah respon keluaran sistem sudah sesuai dengan spesifikasi atau belum. Adapun spesifikasi kontrol yang diinginkan adalah sebagai berikut:

- *Error steady state* (e_{ss})

Error steady state merupakan ukuran dari akurasi sistem kontrol dalam pelacakan masukan perintah atau menyaring gangguan dalam bentuk perubahan beban. Seperti disebutkan sebelumnya, kesalahan tergantung pada jenis sistem dan input, yang meliputi sinyal perintah dan gangguan (A. Frank D'Souza, 1988).

Error steady state yang diinginkan pada respon keluaran sistem nantinya yaitu tidak lebih dari 5% dari nilai akhir (toleransi 5%).

- *Maximum overshoot* (M_p)

Maximum overshoot adalah besaran maksimum dimana respon sistem melebihi nilai keadaan tunaknya. Jadi merupakan amplitudo puncak pertama. *Maximum overshoot* sendiri sering kali ditulis sebagai persentase dari nilai keadaan tunak (W. Bolton, 2004).

Maximum overshoot yang diinginkan pada respon keluaran sistem nantinya yaitu tidak lebih dari 10% (range 0% - 10%).

- *Settling time* (t_s)

Settling time merupakan ukuran dari waktu yang diperlukan oleh osilasi respon sistem untuk ‘menghilang’. Lebih spesifiknya, *settling time* adalah waktu yang diperlukan oleh respon sistem untuk jatuh menuju suatu nilai tertentu dan kemudian tetap berada pada nilai ini (W. Bolton, 2004).

Settling time yang diinginkan pada respon keluaran sistem nantinya yaitu tidak lebih dari 2 jam ($t_s < 2$ jam).

4.5 Prinsip Kerja Alat

Cara kerja alat ini yaitu sebagai berikut:

1. Dimulai dari suhu air normal di dalam tangki evaporator oleh sensor suhu *K Type Thermocouple*, yaitu sekitar 24°C - 26°C . *K Type Thermocouple* sebagai sensor suhu di dalam tangki evaporator.
2. Dengan waktu bersamaan, tekanan udara normal (ketinggian di Malang = ± 540 mdpl) oleh sensor tekanan MPX5100AP, yaitu $\pm 70,6$ cmHg atau $\pm 0,93$ atm. MPX5100AP sebagai sensor tekanan di dalam tangki evaporator dan kondensator.
3. Setelah itu data suhu dan tekanan diolah di dalam Arduino Mega 2560, sehingga nantinya Arduino akan mengeluarkan sinyal terkontrol berupa PWM untuk mengendalikan elemen pemanas.
4. Suhu dijaga agar bisa tetap berada pada 62°C sesuai dengan Gambar 2.3.

4.6 Pemilihan Perangkat Keras (*Hardware*)

Pemilihan perangkat keras terbagi beberapa bagian, diantaranya:

1. Pemilihan elemen pemanas.
2. Pemilihan rangkaian *dimmer* AC.
3. Pemilihan modul *thermocouple amplifier* MAX6675.
4. Pemilihan rangkaian *filter* sensor tekanan MPX5100AP.
5. Modul Arduino Mega 2560.

4.6.1 Pemilihan Elemen Pemanas

Pemilihan dan penempatan elemen pemanas ini didasarkan pada penggunaannya yang dipasang pada jaket luar, yang mana di dalam jaket tersebut berisi minyak goreng sebanyak 15 liter sebagai media pemanasnya.

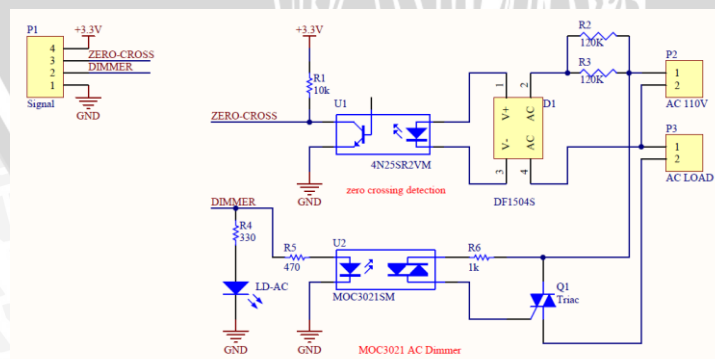
Selain penempatan, harus diketahui juga susunan dan spesifikasi elemen pemanas sebagai ukuran berapa buah elemen pemanas yang akan dipakai dan digunakan. Susunan dan spesifikasinya antara lain, ditempatkan dua buah disebelah kanan dan dua buah disebelah kiri, yang masing elemen pemanas mempunyai daya 300 W, jadi total keseluruhan daya 4 buah elemen pemanas adalah 1200 W. Gambar 4.3 menunjukkan elemen pemanas yang akan digunakan.



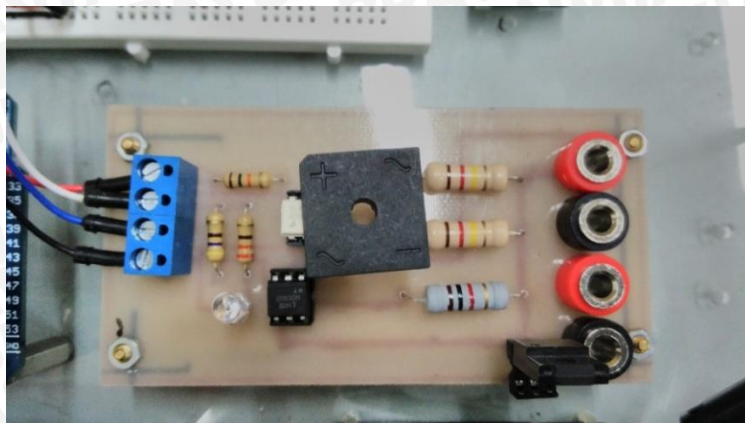
Gambar 4.3 Elemen Pemanas pada Jacket Tabung Evaporator (Perancangan)

4.6.2 Pemilihan Rangkaian *Dimmer* AC

Rangkaian *dimmer* AC dipilih untuk digunakan dalam mengendalikan besarnya arus yang melewati elemen pemanas yang ber-catu daya AC. Di dalam rangkaian *dimmer* AC terdapat beberapa bagian, yaitu rangkaian *Zero Cross Detector* dan rangkaian pemicuan gate TRIAC. Gambar 4.4 menunjukkan skema rangkaian *dimmer* AC, sedangkan rangkaian yang akan digunakan ditunjukkan dalam Gambar 4.5.



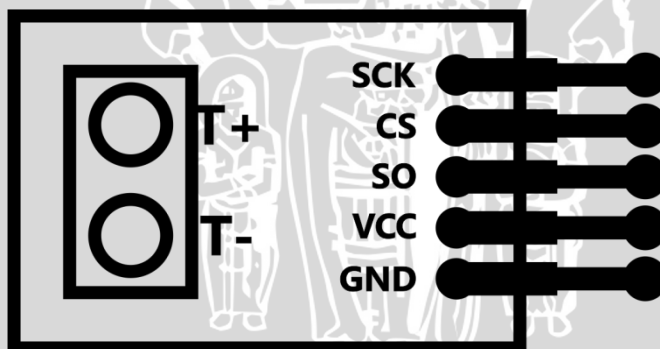
Gambar 4.4 Skema Rangkaian *Dimmer* AC (Firsyari, A. S., 2014)



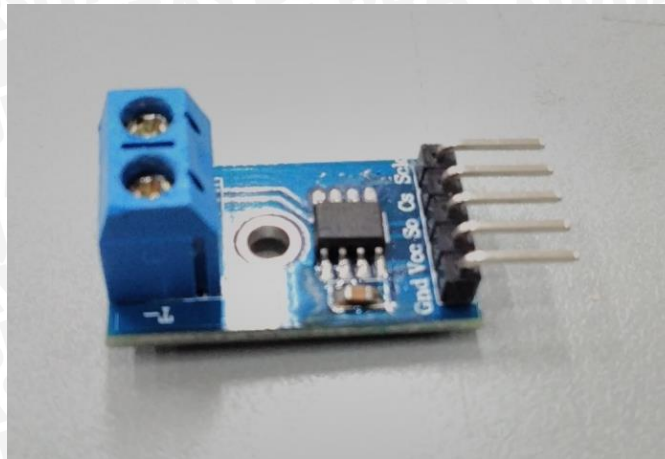
Gambar 4.5 Modul Rangkaian Dimmer AC (Perancangan)

4.6.3 Pemilihan Modul *Thermocouple Amplifier* MAX6675

Modul *Thermocouple Amplifier* MAX6675 dipilih karena kebutuhan untuk digunakan sebagai penguat dan ADC dari sensor suhu *K Type Thermocouple*. Modul ini memiliki dua *input* yang merupakan *output* dari sensor suhu yang nantinya dikeluarkan sebagai data-data digital dalam 5 pin, yaitu VCC, SCK, CS, SO, dan NC. Gambar 4.6 menunjukkan skema modul *Thermocouple Amplifier* MAX6675, sedangkan modul rangkaian ditunjukkan dalam Gambar 4.7.



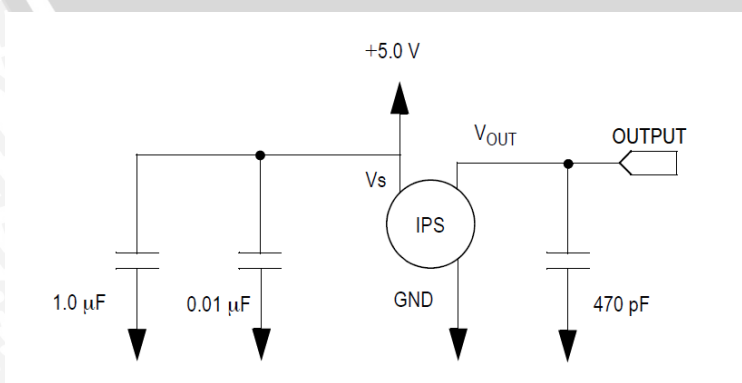
Gambar 4.6 Skema Modul *Thermocouple Amplifier* MAX6675 (Datasheet)



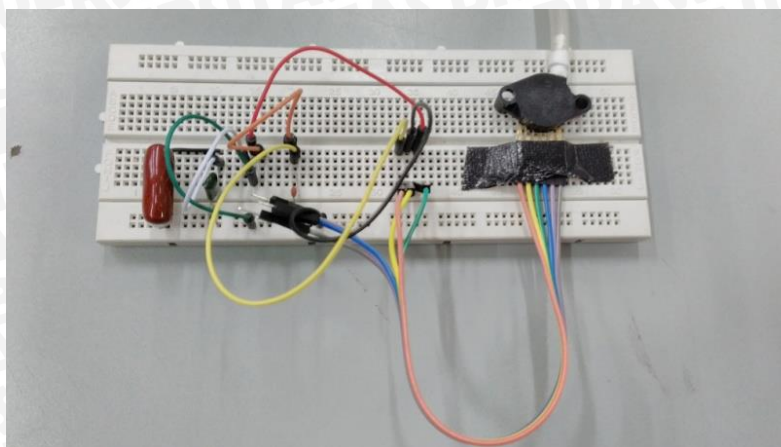
Gambar 4.7 Modul Rangkaian *Thermocouple Amplifier* MAX6675 (Perancangan)

4.6.4 Pemilihan Rangkaian *Filter* Sensor Tekanan MPX5100AP

Keluaran sensor tekanan MPX510AP rentan terhadap *noise* sehingga perlu ditambahkan *filter* untuk mengurangi *noise* yang terjadi. *Noise* dapat disebabkan oleh komponen lain. Komponen yang dapat digunakan sebagai *filter* sensor adalah komponen *decoupling capacitor*. Karena tegangan *power supply* tetap, perubahan daya akan mengakibatkan perubahan arus dan *power supply* harus mampu melayani perubahan arus tersebut. Saat ada perubahan arus tiba-tiba, *power supply* tidak bisa merespon dengan cepat. Akibatnya tegangan pada komponen yang bersangkutan akan drop sebentar sebelum mendapat respon dari *power supply*. Disinilah fungsi *decoupling capacitor* yang dapat menyimpan daya. Kapasitor menyimpan sedikit daya dari *power supply* tetapi dapat merespon perubahan arus dengan cepat. Besarnya nilai kapasitor didasarkan pada *datasheet* sensor, yakni bernilai 1 μF , 0,01 μF , dan 470 pF. Gambar 4.8 menunjukkan skema rangkaian *filter* sensor tekanan MPX5100AP dan modul rangkaian ditunjukkan dalam Gambar 4.9.



Gambar 4.8 Skema Rangkaian *Filter* Sensor Tekanan (Perancangan)



Gambar 4.9 Rangkaian *Filter Sensor Tekanan* (Perancangan)

4.6.5 Modul Arduino Mega 2560

Pada sistem pengendalian suhu ini digunakan Arduino Mega 2560 sebagai pengolah data dalam proses pengendalian elemen pemanas.

Arduino Mega adalah *board* mikrokontroler berbasis ATmega1280. Memiliki 54 pin *input* dan *output* digital, dimana 14 pin diantara pin tersebut dapat digunakan sebagai *output* PWM dan 16 pin *input* analog, 16 MHz osilator kristal, koneksi USB, jack power, ICSP *header*, dan tombol reset. Untuk mendukung mikrokontroler agar dapat digunakan, cukup hanya menghubungkan *Board* Arduino Mega ke komputer dengan menggunakan kabel USB atau dengan adaptor AC-DC untuk menjalankannya.

Secara fisik, ukuran Arduino Mega hampir kurang lebih 2 kali lebih besar dari Arduino Uno, ini untuk mengakomodasi lebih banyaknya pin Digital dan Analog pada board Arduino Mega tersebut. Tampilan Arduino Mega dapat dilihat dalam Gambar 4.10.



Gambar 4.10 Modul Arduino Mega 2560 (www.store.arduino.cc)

Penggunaan pin pada Arduino Mega 2560 disebutkan dalam Tabel 4.1.

Tabel 4.1 Penggunaan Pin Pada Arduino Mega 2560 (Perancangan)

No	Pin	Fungsi
1	8	LCD1 & LCD2 Pin RS
2	9	LCD1 Pin Enable
3	10	LCD2 Pin Enable
4	4	LCD1 & LCD2 Pin D4
5	5	LCD1 & LCD2 Pin D5
6	6	LCD1 & LCD2 Pin D6
7	7	LCD1 & LCD2 Pin D7
8	5 V	Jalur masukan catu daya 5 V
9	GND	Jalur masukan ground
10	2	Jalur masukan <i>Zero Cross Interrupt</i>
11	3	Jalur keluaran untuk rangkaian <i>Dimmer AC</i>
12	45	Jalur ground untuk MAX6675
13	47	Jalur VCC untuk MAX6675
14	49	Jalur SO untuk MAX6675
15	51	Jalur CS untuk MAX6675
16	53	Jalur SCK untuk MAX6675
17	A0	Jalur masukan Sensor Tekanan MPX5100AP

4.7 Perancangan Perangkat Lunak (Software)

Perancangan perangkat lunak terbagi menjadi 2 bagian, yaitu:

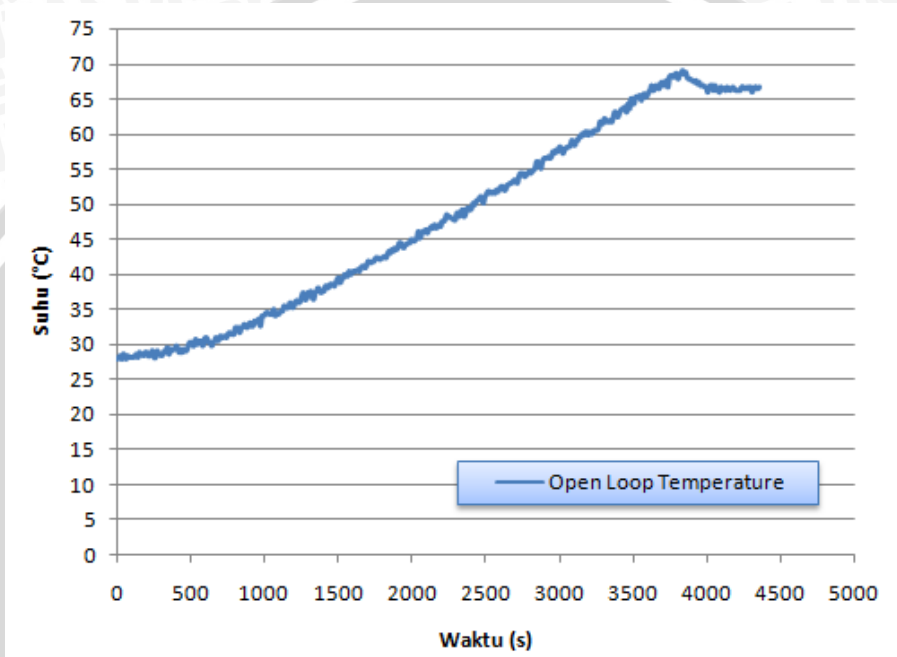
1. Perancangan Algoritma Logika *Fuzzy*.
2. Perancangan *Flowchart*.

4.7.1 Perancangan Algoritma Logika *Fuzzy*

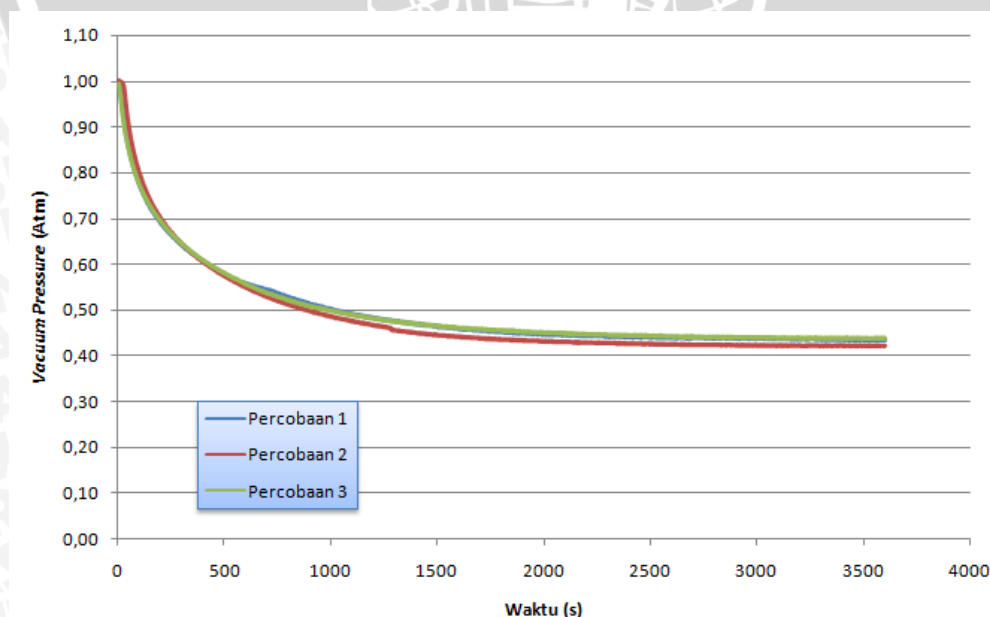
4.7.1.1 *Fuzzy Knowledge-Base*

Fuzzy knowledge-base merupakan dasar pengetahuan dari apa yang nanti akan dirancang pada Algoritma Logika *Fuzzy*. Dasar pengetahuan ini sangat penting dikarenakan *Fuzzy* merupakan sistem

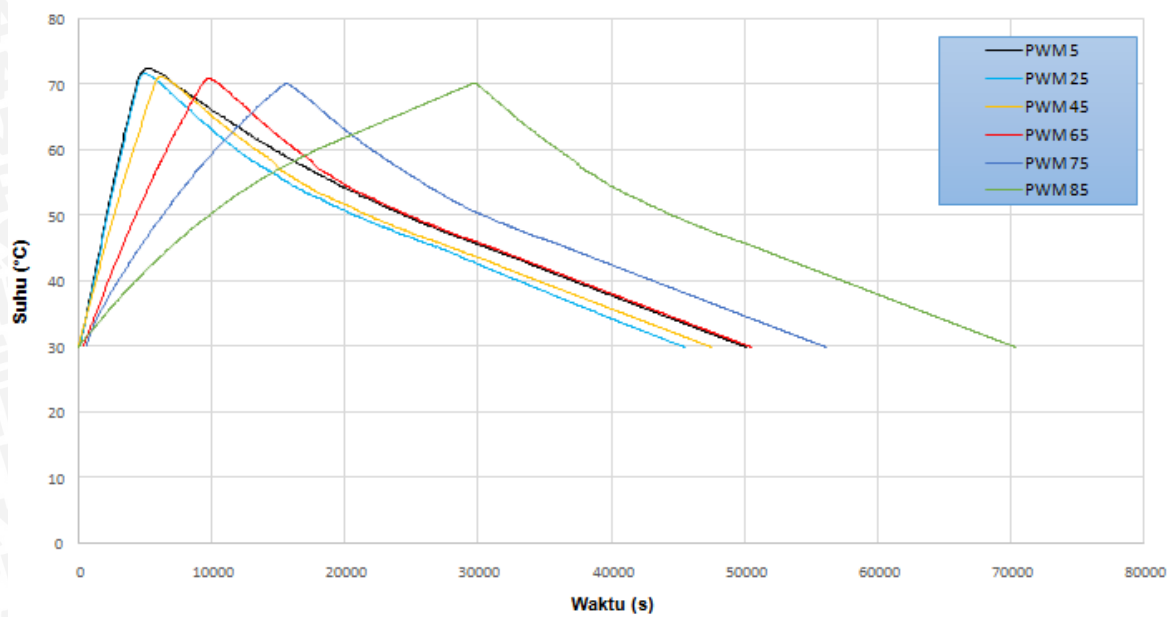
kontrol dengan basis penalaran yang membutuhkan adanya pengetahuan tentang *plant* yang di kontrol dan aktuator, serta variabel yang mempengaruhi sistem. Gambar 4.11 menunjukkan perilaku dari *open loop* dari keluaran suhu dan Gambar 4.12 menunjukkan perilaku *open loop* dari *disturbance* tekanan vakum. Sedangkan perilaku aktuator (perubahan PWM) terhadap suhu keluaran sistem ditunjukkan dalam Gambar 4.13.



Gambar 4.11 Grafik *Open Loop* Respon Keluaran Suhu (Perancangan)



Gambar 4.12 Grafik *Open Loop* Respon Keluaran *Disturbance* Tekanan Vakum (Perancangan)



Gambar 4.13 Grafik Respon Keluaran Suhu terhadap Perubahan PWM (Perancangan)

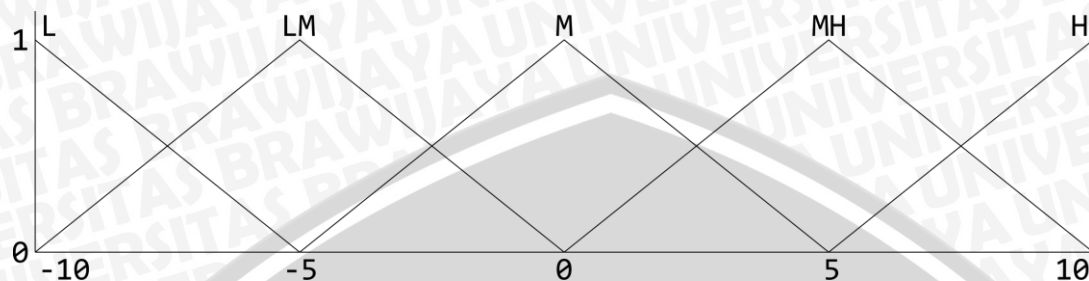
4.7.1.2 Variabel Masukan dan Keluaran

Algoritma Logika Fuzzy yang dibuat di skripsi ini mempunyai dua *crisp input* yaitu *temperature error* dan *vacuum pressure*, serta satu *crisp output* yaitu PWM. *Temperature error* adalah selisih antara *setpoint* dan umpan balik keluaran plant yang dibaca oleh sensor *K Type Thermocouple*. Sedangkan *vacuum pressure input* merupakan suatu *disturbance* yang berpengaruh langsung ke dalam sistem. Sebelum masuk ke dalam Algoritma Logika Fuzzy, *vacuum pressure input* dibaca dahulu oleh sensor MPX5100AP, sehingga mengeluarkan sinyal digital yang nantinya akan diubah menjadi nilai tekanan oleh display (*serial monitor*).

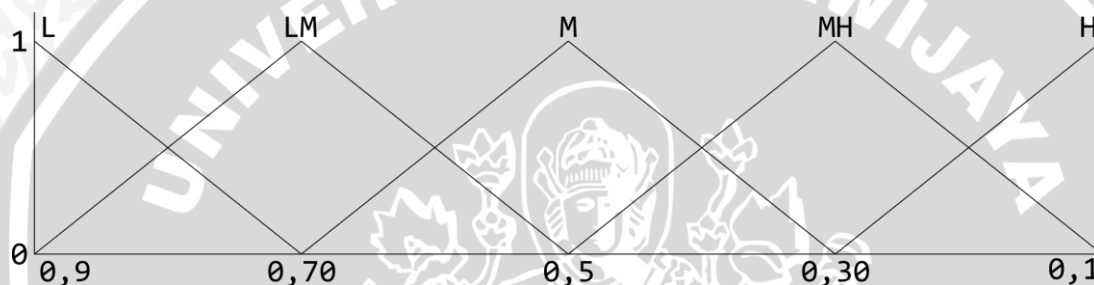
4.7.1.3 Fungsi Keanggotaan Masukan dan Keluaran

Fungsi keanggotaan dari *temperature error* dan *vacuum pressure* terdiri atas lima label, yaitu *Low (L)*, *Low-Medium (LM)*, *Medium (M)*, *Medium-High (MH)*, dan *High (H)*. Label masing-masing fungsi keanggotaan berjumlah lima dengan tujuan agar *rule* (aturan) yang dihasilkan semakin banyak, karena semakin banyak aturan maka semakin banyak kemungkinan kondisi yang tercipta dan

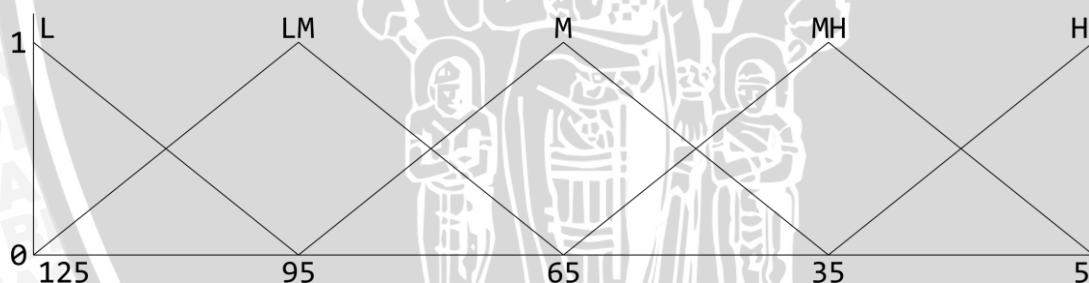
semakin banyak pula aksi kontrol yang akan dipilih. Gambar 4.14 dan 4.15 merupakan gambar fungsi keanggotaan masukan yang direncanakan. Sedangkan Gambar 4.16 merupakan tampilan dari fungsi keanggotaan keluaran yang nantinya akan digunakan.



Gambar 4.14 Fungsi Keanggotaan Masukan *Temperature Error* (Perancangan)



Gambar 4.15 Fungsi Keanggotaan Masukan *Vacuum Pressure* (Perancangan)



Gambar 4.16 Fungsi Keanggotaan Keluaran PWM (Perancangan)

4.7.1.4 Perancangan Aturan Fuzzy

Aturan fuzzy (*rule*) digunakan sebagai penentu keluaran dari fuzzifikasi yang akan diolah dalam proses defuzzifikasi, dengan jumlah 25 macam *rule*. *Rule* tersebut didapatkan dari 5 buah label dalam fungsi keanggotaan masukan *temperature error* dan *vacuum pressure*. Hubungan antara *temperature error* dan *vacuum pressure* terhadap nilai PWM dijabarkan sebagai berikut:

- *Temperature error* = *setpoint* – umpan balik.

- Suhu semakin tinggi (panas), maka tekanan vakum semakin besar.
- Tekanan vakum semakin kecil, maka titik didih semakin rendah.
- Label masukan *temperatur error* semakin besar (bernilai positif (+)), maka label keluaran (PWM) semakin besar.
- Label masukan *vacuum pressure* semakin besar, maka label keluaran (PWM) semakin besar.
- Label masukan *temperature error* dan *vacuum pressure* semakin besar (bernilai positif (+) untuk *temperature error*), maka label keluaran (PWM) juga akan semakin besar.

Dengan acuan diatas, diperoleh perancangan aturan *fuzzy* dalam Tabel 4.2 dibawah ini.

Tabel 4.2 Fuzzy Rule (Perancangan)

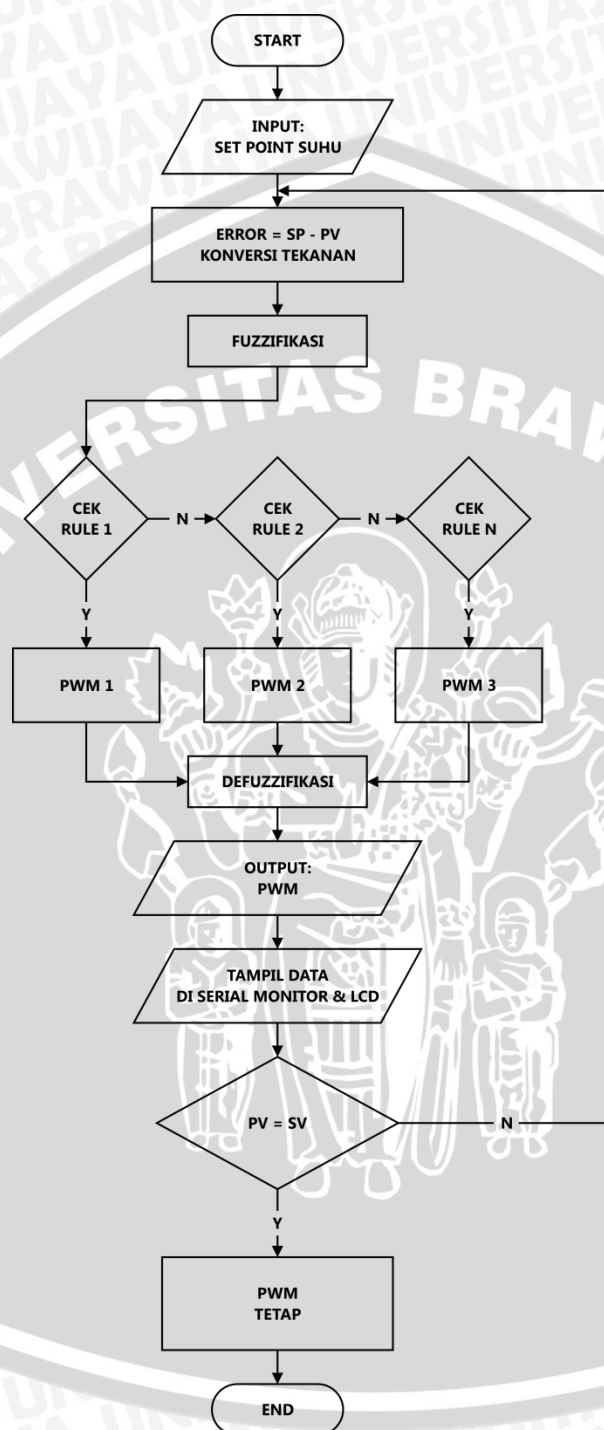
FUZZY RULE		Error				
		L	LM	M	MH	H
Vacuum Pressure	L	L	L	L	LM	M
	LM	L	LM	LM	M	MH
	M	L	LM	M	MH	H
	MH	LM	M	MH	MH	H
	H	M	MH	H	H	H

4.7.1.5 Metode Inferensi dan Defuzzifikasi

Metode inferensi fuzzy yang direncanakan menggunakan *Max-Min*. Setelah dari *crisp input* diubah menjadi *fuzzy* melalui proses fuzzifikasi, maka untuk melihat keluaran dari Algoritma Logika Fuzzy ini nilai-nilai *fuzzy* tersebut harus diubah menjadi bentuk *crisp* lagi menggunakan proses yang disebut defuzzifikasi. Hasil dari defuzzifikasi ini akan digunakan sebagai pengatur PWM. Metode defuzzifikasi yang direncanakan adalah menggunakan *Weighted Average (WA)*.

4.7.2 Perancangan Flowchart

Gambar 4.17 merupakan *flowchart* sistem secara keseluruhan yang juga berkaitan dengan *flowchart* program pada *software* Arduino ERW 1.6.5.



Gambar 4.17 Flowchart Sistem Keseluruhan (Perancangan)



BAB V

PENGUJIAN DAN ANALISIS SISTEM

Pengujian sistem disini bertujuan untuk mengetahui apakah alat yang dibuat bisa berfungsi dengan perancangannya. Pengujian sistem ini diantaranya adalah pengujian setiap blok serta pengujian sistem secara keseluruhan. Pengujian dibagi menjadi beberapa bagian, yaitu:

1. Pengujian sensor suhu *K Type Thermocouple*
2. Pengujian modul *Thermocouple Amplifier MAX6675*
3. Pengujian sensor tekanan MPX5100AP
4. Pengujian LCD
5. Pengujian penyalan TRIAC
6. Pengujian rangkaian *Dimmer AC*
7. Pengujian sistem keseluruhan

5.1 Pengujian Sensor Suhu *K Type Thermocouple*

5.1.1 Tujuan

Mengetahui kemampuan pembacaan sensor *K Type Thermocouple* terhadap perubahan suhu dengan melihat perubahan keluaran tegangan dari sensor *K Type Thermocouple*.

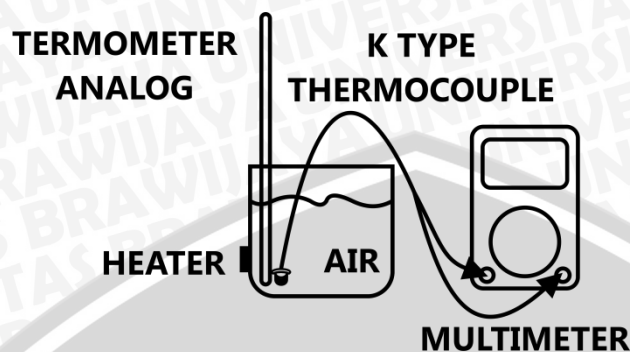
5.1.2 Peralatan yang digunakan

- Sensor *K Type Thermocouple*
- Air
- Termometer Analog
- *Heater*
- Multimeter

5.1.3 Prosedur pengujian

1. Menyusun rangkaian seperti dalam Gambar 5.1.
2. Menghubungkan catu daya *heater* dengan sumber AC.

3. Mengamati dan mencatat setiap perubahan keluaran tegangan sensor *K Type Thermocouple* menggunakan multimeter terhadap perubahan suhu berdasarkan suhu termometer acuan.



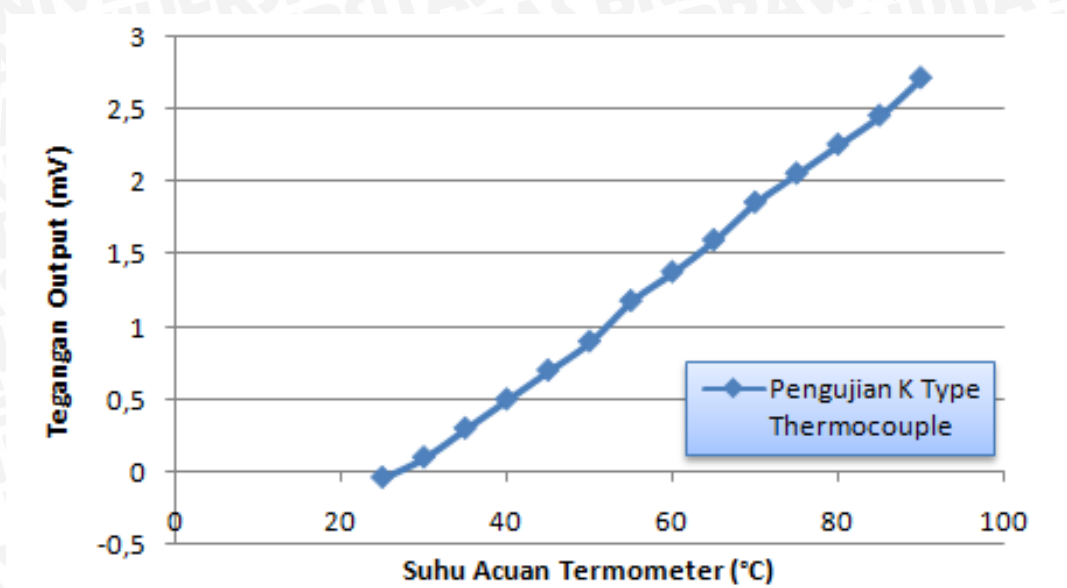
Gambar 5.1 Pengujian Sensor *K Type Thermocouple* (Perancangan)

5.1.4 Hasil pengujian

Hasil pengujian *K Type Thermocouple* ditunjukkan dalam Tabel 5.1. Didapatkan grafik *linearitas* antara suhu dengan tegangan seperti yang ditunjukkan dalam Gambar 5.2.

Tabel 5.1 Hasil Pengujian Sensor *K Type Thermocouple* (Perancangan)

No.	Suhu Pembacaan Termometer (°C)	Tegangan Pembacaan Sensor (mV)
1.	25,00	-0,04
2.	30,00	0,10
3.	35,00	0,30
4.	40,00	0,50
5.	45,00	0,70
6.	50,00	0,90
7.	55,00	1,18
8.	60,00	1,38
9.	65,00	1,60
10.	70,00	1,86
11.	75,00	2,06
12.	80,00	2,26
13.	85,00	2,46
14.	90,00	2,72



Gambar 5.2 Grafik Hubungan Tegangan K Type Thermocouple dengan Suhu (Perancangan)

5.2 Pengujian Modul *Thermocouple Amplifier* MAX6675

5.2.1 Tujuan

Mengetahui kinerja dari modul *Thermocouple Amplifier* MAX6675 terhadap perubahan suhu dengan melihat hasil pembacaan suhu melalui *Serial Monitor* dalam *software* Arduino ERW 1.6.5.

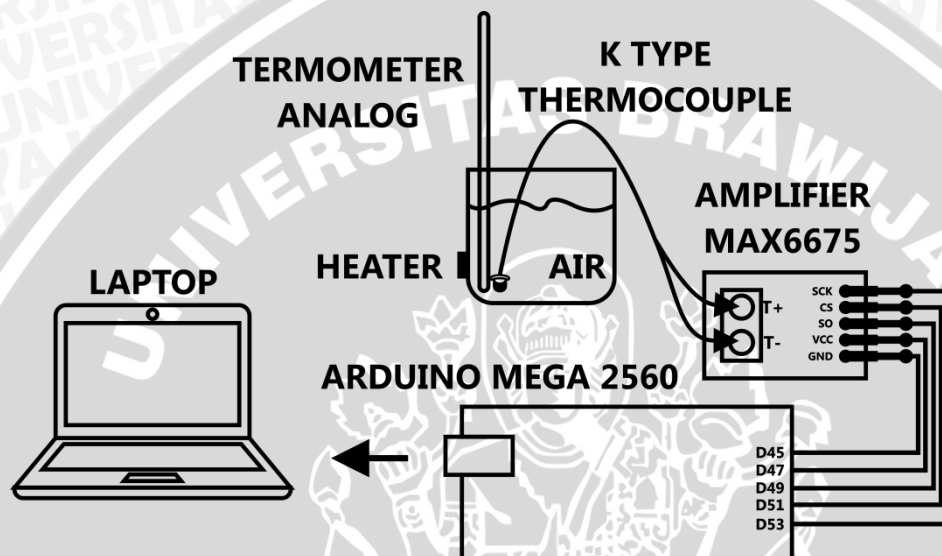
5.2.2 Peralatan yang digunakan

- Sensor K Type Thermocouple
- Modul *Thermocouple Amplifier* MAX6675
- Air
- Termometer Analog
- Heater
- Arduino Mega 2560
- Laptop

5.2.3 Prosedur pengujian

1. Menyusun rangkaian pengujian seperti dalam Gambar 5.3.
2. Menghubungkan K Type Thermocouple dengan modul *Thermocouple Amplifier* MAX6675.

3. Menghubungkan port *Thermocouple Amplifier* MAX6675 dengan pin digital dari Arduino Mega 2560.
4. Menghubungkan Arduino Mega 2560 dengan Laptop.
5. Memanaskan air dalam *heater* dengan *K Type Thermocouple* dan siapkan termometer analog.
6. Hidupkan *Serial Monitor* pada *software* Arduino ERW 1.6.5.
7. Amati dan catat setiap perubahan suhu pada termometer dan hasil pembacaan suhu pada *Serial Monitor*.



Gambar 5.3 Pengujian Modul *Thermocouple Amplifier* MAX6675 (Perancangan)

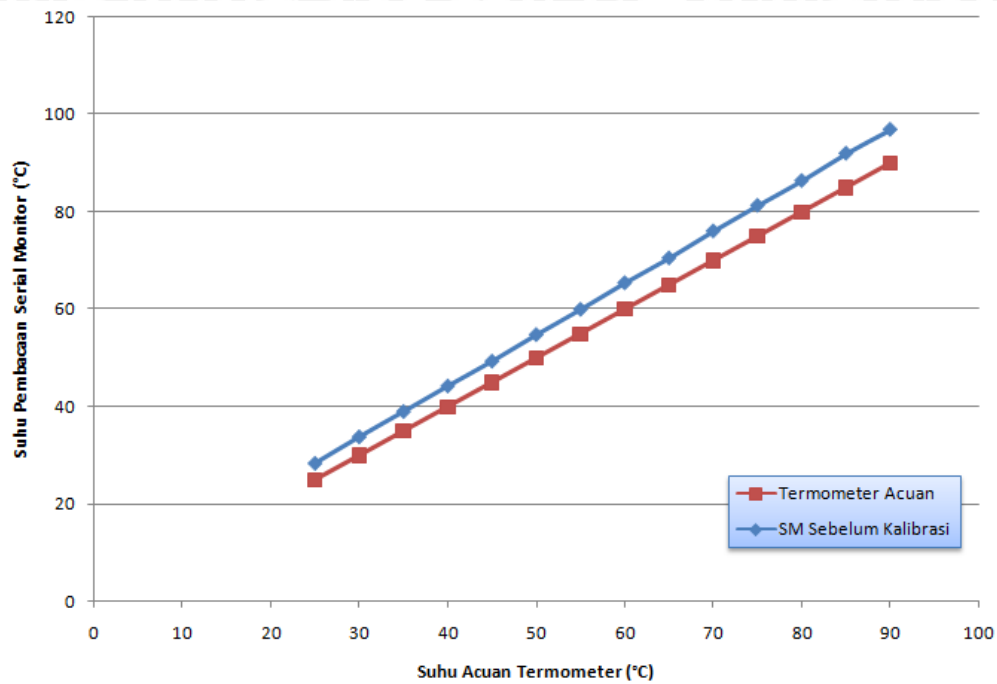
5.2.4 Hasil pengujian

Hasil pengujian modul *Thermocouple Amplifier* MAX6675 ditunjukkan dalam Tabel 5.2.

Tabel 5.2 Hasil Pengujian Modul *Thermocouple Amplifier* MAX6675 (Perancangan)

No.	Suhu Pembacaan Termometer (°C)	Suhu Pembacaan Serial Monitor (°C)	Error (%)
1.	25,00	28,30	13,20
2.	30,00	33,75	12,50
3.	35,00	38,95	11,29
4.	40,00	44,20	10,50
5.	45,00	49,30	9,56
6.	50,00	54,75	9,50
7.	55,00	59,95	9,00
8.	60,00	65,40	9,00
9.	65,00	70,50	8,46
10.	70,00	76,05	8,64
11.	75,00	81,30	8,40
12.	80,00	86,40	8,00
13.	85,00	92,00	8,24
14.	90,00	96,90	7,67
Rata-rata			9,567687588

Berdasarkan hasil pengujian dalam Tabel 5.2, *error* antara suhu pembacaan termometer dengan suhu pembacaan *serial monitor* masih sangat besar yaitu 9,567687588%. Grafik perbandingan suhu hasil pembacaan termometer acuan dengan *serial monitor* dapat dilihat dalam Gambar 5.4. Pada grafik tersebut terlihat bahwa selisih yang timbul antara kedua pembacaan suhu cenderung membentuk garis linear. Sehingga untuk mengkompensasi *error* pembacaan suhu, dapat menggunakan cara kalibrasi. Rumus umum untuk kalibrasi terdapat dalam Persamaan (5-1) dan Persamaan (5-2), serta hasil akhir rumus kalibrasi ditunjukkan dalam Persamaan (5-3).



Gambar 5.4 Grafik Perbandingan Suhu Hasil Pembacaan Termometer Acuan dengan *Serial Monitor* (Perancangan)

Rumus umum kalibrasi:

$$\sum x_i a_0 + \sum x_i^2 a_1 = \sum x_i y_i \quad (5-1)$$

$$f = a_0 + a_1 x \quad (5-2)$$

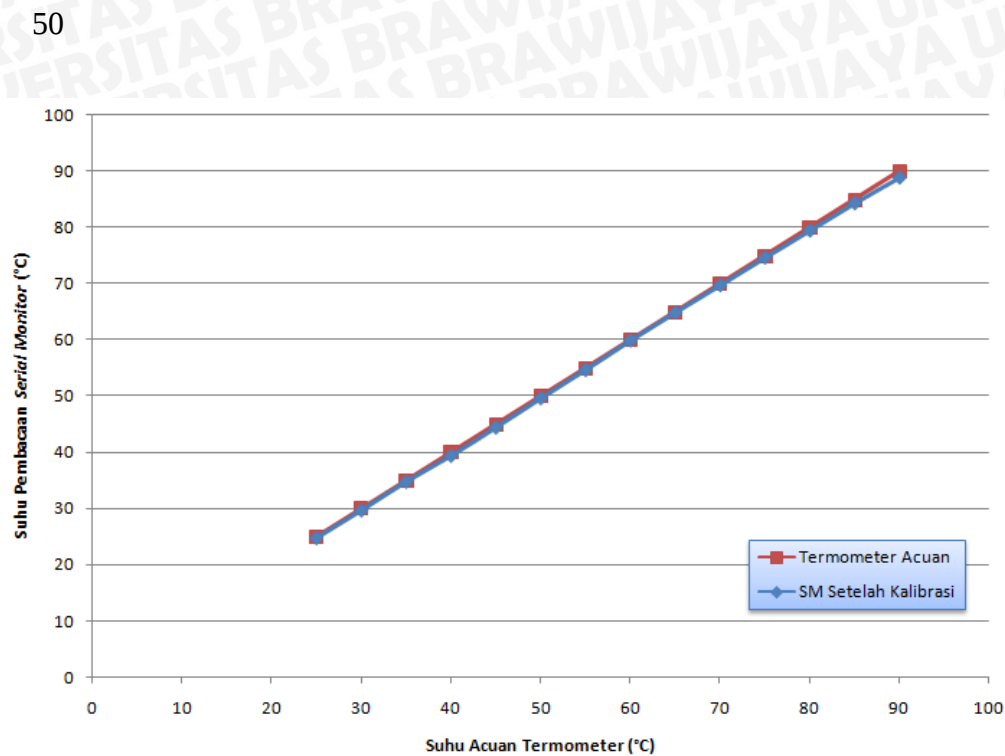
$$f = -1,759384283 + 0,945179584x \quad (5-3)$$

Kompensasi *error* pembacaan suhu dilakukan pada program di dalam Arduino Mega 2560. Sehingga menghasilkan data pengujian yang ditunjukkan dalam Tabel 5.3.

Tabel 5.3 Hasil Pengujian Modul *Thermocouple Amplifier* MAX6675 Setelah Kalibrasi (Perancangan)

No.	Suhu Pembacaan Termometer (°C)	Suhu Pembacaan Serial Monitor (°C)	Error (%)
1.	25,00	24,732	1,07
2.	30,00	29,662	1,13
3.	35,00	34,766	0,67
4.	40,00	39,388	1,53
5.	45,00	44,422	1,28
6.	50,00	49,626	0,75
7.	55,00	54,670	0,60
8.	60,00	59,912	0,15
9.	65,00	64,878	0,19
10.	70,00	69,746	0,36
11.	75,00	74,660	0,45
12.	80,00	79,478	0,65
13.	85,00	84,348	0,77
14.	90,00	88,908	1,21
Rata-rata			0,772366011

Berdasarkan Tabel 5.3, hasil pengujian modul *Thermocouple Amplifier* MAX6675 sensor *K Type Thermocouple* menghasilkan suhu pembacaan *serial monitor* yang mendekati suhu acuan dengan *error* rata-rata sebesar 0,772366011% seperti yang ditunjukkan dalam Gambar 5.5, sehingga kalibrasi yang dilakukan telah sesuai dan modul *Thermocouple Amplifier* MAX6675 dapat bekerja dengan baik.



Gambar 5.5 Grafik Perbandingan Suhu Hasil Pembacaan Termometer Acuan dengan *Serial Monitor* Setelah Kalibrasi (Perancangan)

5.3 Pengujian Sensor Tekanan MPX5100AP

5.3.1 Tujuan

Mengetahui kemampuan pembacaan sensor MPX5100AP terhadap perubahan tekanan dengan melihat hasil pembacaan tekanan melalui *Serial Monitor* dalam *software* Arduino ERW 1.6.5.

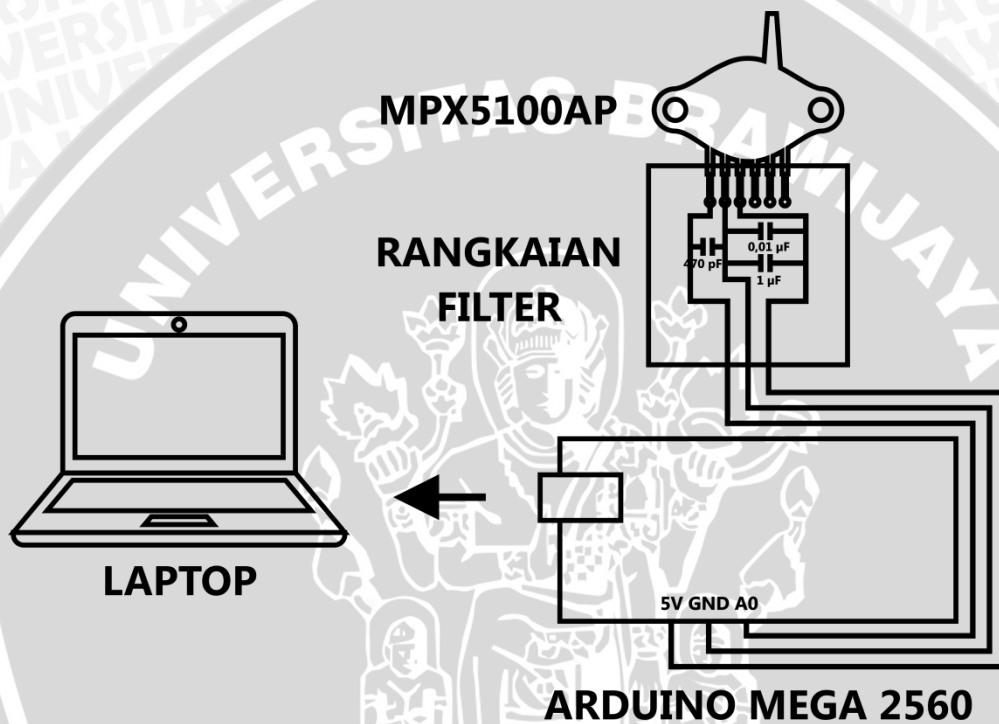
5.3.2 Peralatan yang digunakan

- Sensor Tekanan MPX5100AP
- Rangkaian *Filter*
- Arduino Mega 2560
- Laptop
- *Pressure Vacuum Gauge*
- Piranti Vakum

5.3.3 Prosedur pengujian

1. Menyusun rangkaian pengujian seperti dalam Gambar 5.6.
2. Menghubungkan MPX5100AP dengan rangkaian *filter*.

3. Menghubungkan rangkaian *filter* (VCC, GND, Vout) dengan pin analog dari Arduino Mega 2560.
4. Menghubungkan Arduino Mega 2560 dengan Laptop.
5. Memompa udara dalam piranti vakum dengan Pompa dan dilihat dengan *Pressure Vacuum Gauge*.
6. Hidupkan *Serial Monitor* pada *software* Arduino ERW 1.6.5.
7. Amati dan catat setiap perubahan tekanan pada *Pressure Vacuum Gauge* dan hasil pembacaan suhu pada *Serial Monitor*.



Gambar 5.6 Pengujian Sensor Tekanan Vakum MPX5100AP (Perancangan)

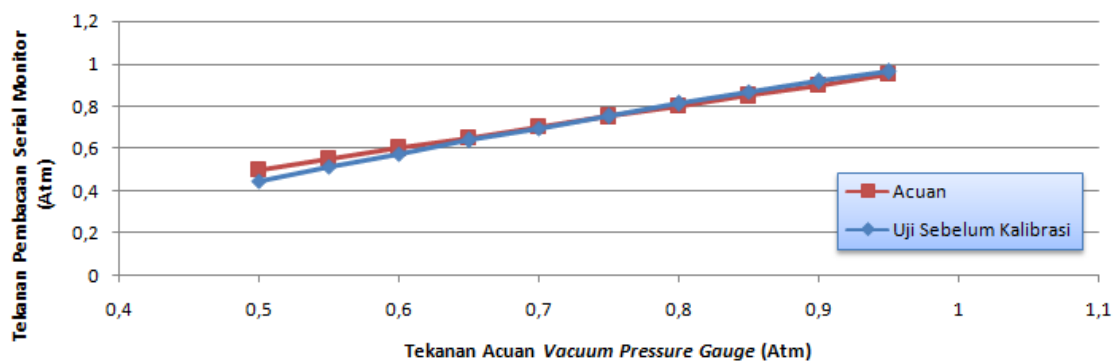
5.3.4 Hasil pengujian

Hasil pengujian sensor tekanan vakum MPX5100AP ditunjukkan dalam Tabel 5.4.

Tabel 5.4 Hasil Pengujian Sensor Tekanan Vakum MPX5100AP (Perancangan)

No.	Tekanan Pembacaan Pressure Gauge (Atm)	Pembacaan ADC Serial Monitor	Tekanan Pembacaan Serial Monitor (Atm)	Error (%)
1.	0,95	756	0,96	1,33
2.	0,90	720	0,92	1,91
3.	0,85	682	0,87	2,16
4.	0,80	638	0,81	1,54
5.	0,75	591	0,75	0,33
6.	0,70	546	0,70	0,64
7.	0,65	500	0,64	2,01
8.	0,60	450	0,57	4,53
9.	0,55	402	0,51	6,89
10.	0,50	350	0,45	10,91
Rata-rata				3,224466647

Berdasarkan hasil pengujian dalam Tabel 5.4, *error* antara tekanan pembacaan *Pressure Vacuum Gauge* dengan tekanan pembacaan *serial monitor* masih sangat besar yaitu 3,224466647 %. Grafik perbandingan tekanan hasil pembacaan *Pressure Vacuum Gauge* acuan dengan *serial monitor* dapat dilihat dalam Gambar 5.7. Pada grafik tersebut terlihat bahwa selisih yang timbul antara kedua pembacaan tekanan cenderung membentuk garis linear. Sehingga untuk mengkompensasi *error* pembacaan tekanan, dapat menggunakan cara kalibrasi. Rumus umum untuk kalibrasi terdapat dalam Persamaan (5-4) dan Persamaan (5-5), serta hasil akhir rumus kalibrasi ditunjukkan dalam Persamaan (5-6).



Gambar 5.7 Grafik Perbandingan Tekanan Hasil Pembacaan Pressure Gauge Acuan dengan Serial Monitor (Perancangan)

Rumus umum kalibrasi:

$$\sum x_i a_0 + \sum x_i^2 a_1 = \sum x_i y_i \quad (5-4)$$

$$f = a_0 + a_1 x \quad (5-5)$$

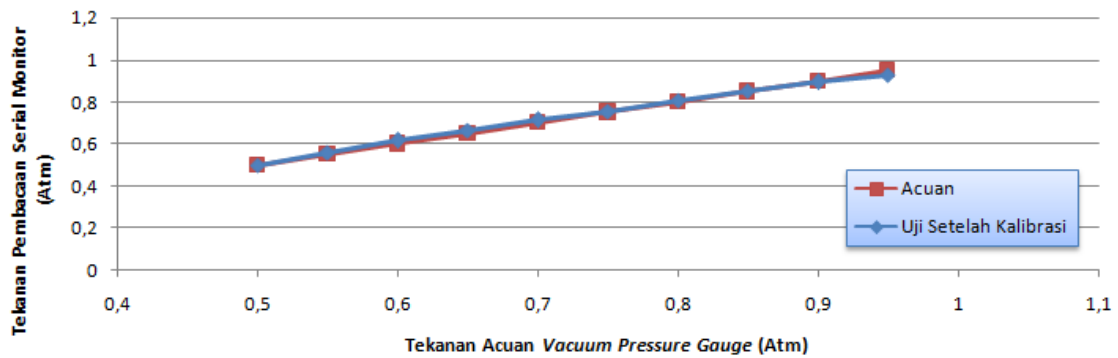
$$f = 0,137413074 + 0,818845619x \quad (5-6)$$

Kompensasi *error* pembacaan tekanan dilakukan pada program di dalam Arduino Mega 2560. Sehingga menghasilkan data pengujian dalam Tabel 5.5.

Tabel 5.5 Hasil Pengujian Sensor Tekanan Vakum MPX5100AP Setelah Kalibrasi (Perancangan)

No.	Tekanan Pembacaan Pressure Gauge (Atm)	Tekanan Pembacaan Serial Monitor (Atm)	Error (%)
1.	0,95	0,93	2,11
2.	0,90	0,89	0,74
3.	0,85	0,85	0,39
4.	0,80	0,81	0,83
5.	0,75	0,76	0,89
6.	0,70	0,71	1,90
7.	0,65	0,67	2,56
8.	0,60	0,62	2,78
9.	0,55	0,56	1,81
10.	0,50	0,50	0,00
Rata-rata			1,402520705

Berdasarkan Tabel 5.5, hasil pengujian sensor tekanan MPX5100AP menghasilkan tekanan pembacaan *serial monitor* yang mendekati suhu acuan dengan *error* rata-rata sebesar 1,402520705 % seperti yang ditunjukkan dalam Gambar 5.8, sehingga kalibrasi yang dilakukan telah sesuai dan MPX5100AP dapat bekerja dengan baik.



Gambar 5.8 Grafik Perbandingan Tekanan Hasil Pembacaan Pressure Gauge Acuan dengan *Serial Monitor* Setelah Kalibrasi (Perancangan)

5.4 Pengujian LCD

5.4.1 Tujuan

Mengetahui kinerja dari *liquid crystal display* (LCD) yang mana akan digunakan untuk menunjukkan waktu dan *error* pada sistem keseluruhan.

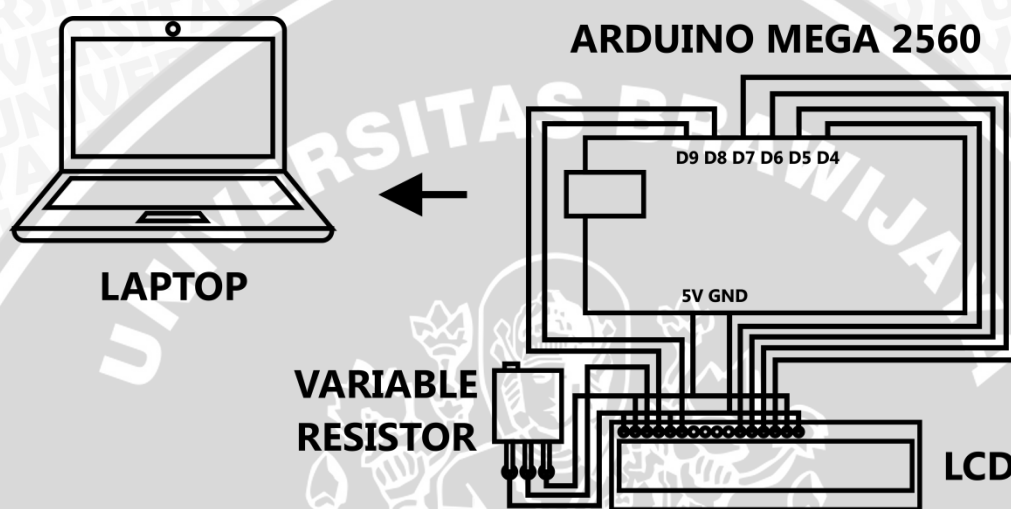
5.4.2 Peralatan yang digunakan

- *Liquid Crystal Display* (LCD)
- Pin Header
- Kabel Male to Female
- Project Board
- 10 K Ω Potensiometer
- Arduino Mega 2560
- Laptop

5.4.3 Prosedur pengujian

1. Menyusun rangkaian seperti dalam Gambar 5.9.
2. Menghubungkan dengan antara pin header yang telah terpasang di *liquid crystal display* (LCD) dan di *project board* dengan pin pada arduino mega 2560.

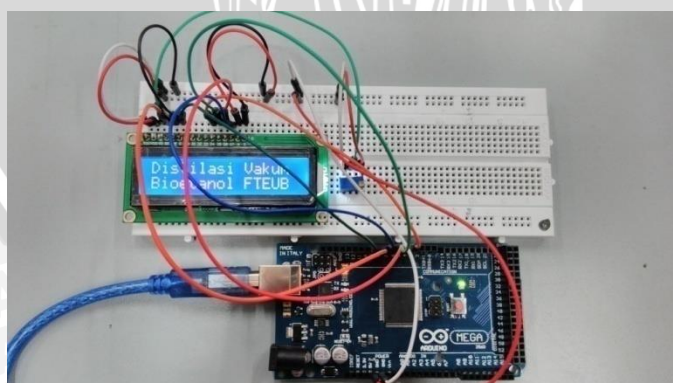
3. Menghubungkan arduino mega 2560 dengan laptop.
4. Membuat dan mengunduh program untuk *liquid crystal display* (LCD) di dalam *software* arduino ERW 1.6.5.
5. Melihat dan mengamati apakah *liquid crystal display* (LCD) sudah tampil sesuai dengan yang diinginkan.
6. Apabila belum terlihat, maka potensiometer harus diputar sampai tulisan yang sudah diprogram dalam *software* arduino ERW 1.6.5 bisa terlihat dengan baik.



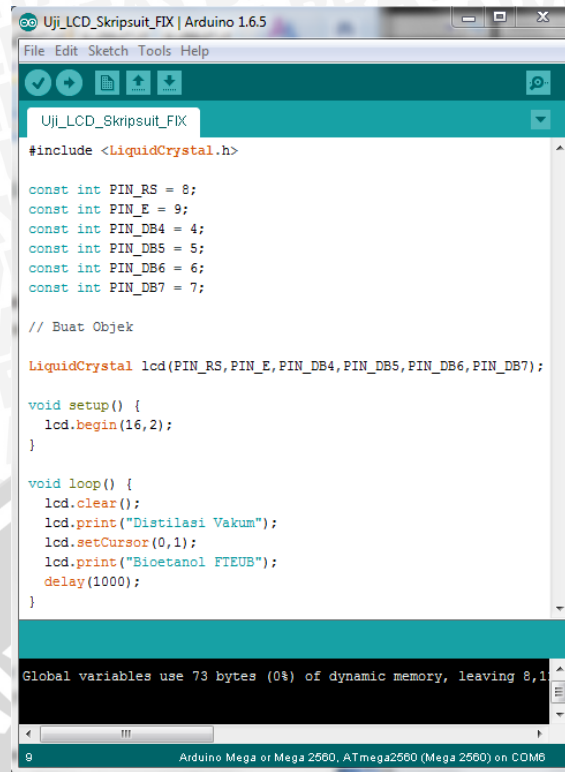
Gambar 5.9 Pengujian *Liquid Crystal Display* (LCD) (Perancangan)

5.4.4 Hasil pengujian

Setelah melakukan prosedur pengujian, maka didapatkan hasil yang ditunjukkan dalam Gambar 5.10, dan *script* pada arduino ERW 1.6.5 dalam Gambar 5.11.



Gambar 5.10 Hasil Pengujian *Liquid Crystal Display* (LCD) (Perancangan)



Gambar 5.11 Script Liquid Crystal Display (LCD) (Perancangan)

5.5 Pengujian Penyalan TRIAC

5.5.1 Tujuan

Pengujian ini dilakukan untuk mengetahui apakah rangkaian yang telah dirancang bisa mengatur gelombang sinusoida tegangan AC.

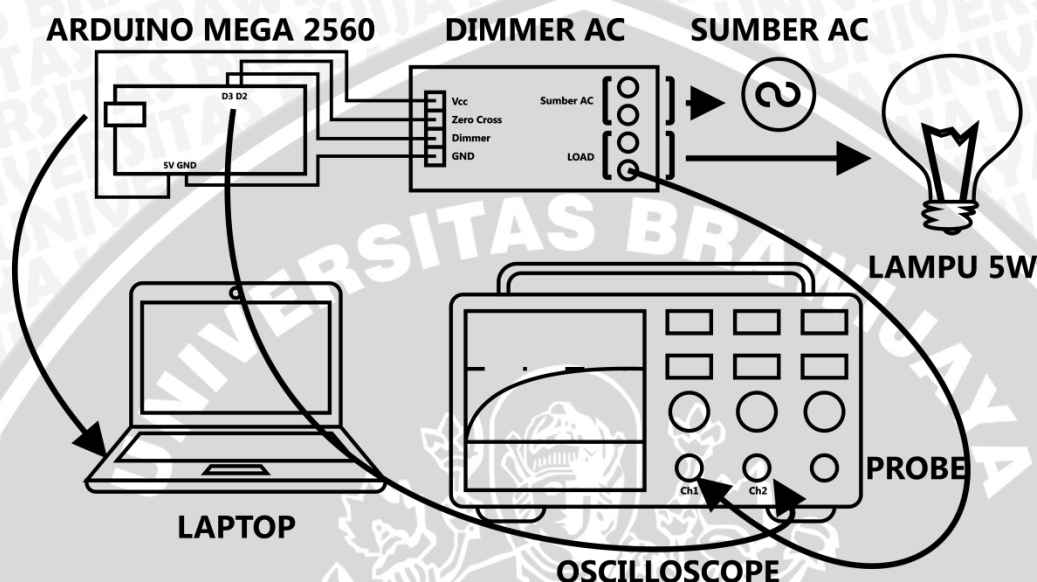
5.5.2 Peralatan yang digunakan

- Rangkaian TRIAC dan MOC3022
- Rangkaian Zero Cross Detector
- Arduino Mega 2560
- Oscilloscope
- Sumber Tegangan AC
- Beban Lampu 5 watt
- Laptop

5.5.3 Prosedur pengujian

1. Menyusun rangkaian seperti dalam Gambar 5.12.
2. Menghubungkan sumber tegangan 5V ke arduino dan ke *optocoupler* 4N25.

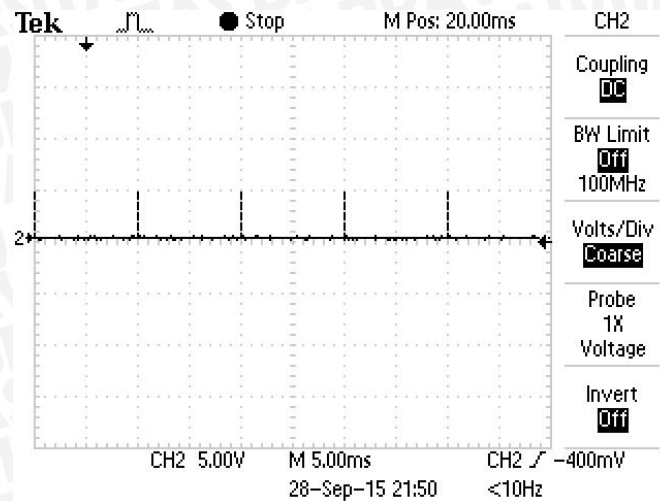
3. Menghubungkan rangkaian ke sumber tegangan AC.
4. Memberi beban berupa lampu 5 watt pada AC LOAD.
5. Menghubungkan keluaran *optocoupler* 4N25 ke pin *interrupt* arduino.
6. Menghubungkan salah satu pin digital I/O dari arduino ke MOC3022.
7. Menghubungkan *probe* 1 *oscilloscope* ke keluaran arduino dan *probe* 2 ke tegangan keluaran TRIAC.



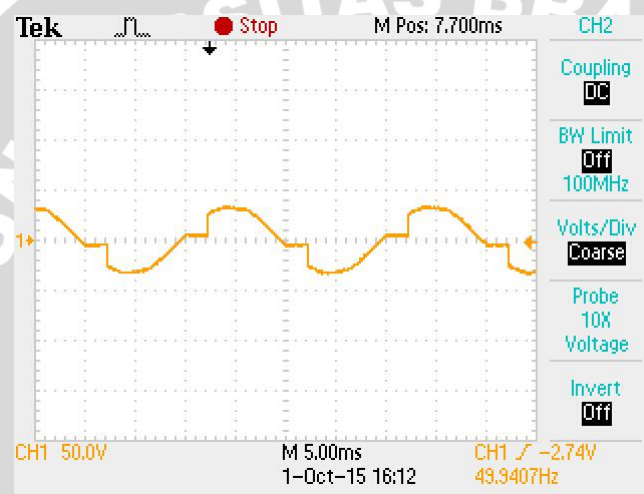
Gambar 5.12 Pengujian Rangkaian Penyalan TRIAC (Perancangan)

5.5.4 Hasil pengujian

Dari pengujian penyalan TRIAC didapatkan hasil pengujian berupa bentuk gelombang keluaran TRIAC sesuai dengan sudut penyalannya. Contoh hasil keluaran gelombang untuk sudut penyalan 45° atau dengan PWM 32 ditunjukkan dalam Gambar 5.13. Hasil pengujian keseluruhan ditunjukkan dalam Tabel 5.6 dan Gambar 5.14.



(a)

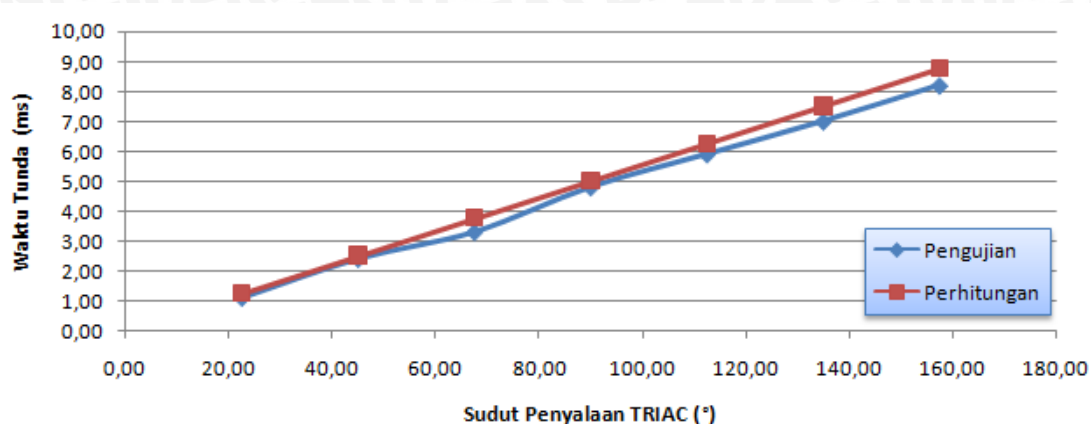


(b)

Gambar 5.13 (a) Bentuk Gelombang Keluaran Arduino, (b) Bentuk Gelombang Keluaran TRIAC, dengan Sudut Penyalaan 45° atau dengan PWM 32 (Perancangan)

Tabel 5.6 Hasil Pengujian Sudut Penyalaan TRIAC (Perancangan)

Data ke-	PWM	Sudut Penyalaan ($^\circ$)	Perhitungan Waktu Tunda (ms)	Pengujian Waktu Tunda (ms)
1	16	22,5	1,25	1,10
2	32	45	2,50	2,40
3	48	67,5	3,75	3,30
4	64	90	5,00	4,80
5	80	112,5	6,25	5,90
6	96	135	7,50	7,00
7	112	157,5	8,75	8,20



Gambar 5.14 Grafik Perbandingan Data Perhitungan dan Pengujian Waktu Tunda TRIAC (Perancangan)

Berdasarkan data hasil pengujian sudut penyalaan TRIAC yang didapatkan dalam Tabel 5.6 terjadi penyimpangan waktu tunda dengan data hasil perhitungan. Berikut 3 contoh perhitungan pengujian dengan sampel data ke-2, ke-4, dan ke-6.

Contoh perhitungan pengujian dengan sampel data ke-2:

$$\begin{aligned}\text{Waktu tunda (ms)} &= \frac{\text{Sudut Penyalaan}}{180^\circ} \times 10 \\ &= \frac{45^\circ}{180^\circ} \times 10 = 2,50 \text{ ms}\end{aligned}$$

Nilai kesalahan yang didapat dari selisih *absolute* perhitungan dan pengujian sebesar:

$$\begin{aligned}\text{Kesalahan (ms)} &= |\text{Perhitungan} - \text{Pengujian}| \\ &= |2,50 - 2,40| = 0,10 \text{ ms} \\ \text{Kesalahan (\%)} &= \frac{\text{Kesalahan}}{\text{Data Perhitungan}} \times 100\% \\ &= \frac{0,10}{2,50} \times 100\% = 4,00\%\end{aligned}$$

Contoh perhitungan pengujian dengan sampel data ke-4:

$$\begin{aligned}\text{Waktu tunda (ms)} &= \frac{\text{Sudut Penyalaan}}{180^\circ} \times 10 \\ &= \frac{90^\circ}{180^\circ} \times 10 = 5,00 \text{ ms}\end{aligned}$$

Nilai kesalahan yang didapat dari selisih *absolute* perhitungan dan pengujian sebesar:

$$\begin{aligned}\text{Kesalahan (ms)} &= |\text{Perhitungan} - \text{Pengujian}| \\ &= |5,00 - 4,80| = 0,20 \text{ ms}\end{aligned}$$

$$\begin{aligned}\text{Kesalahan (\%)} &= \frac{\text{Kesalahan}}{\text{Data Perhitungan}} \times 100\% \\ &= \frac{0,20}{5,00} \times 100\% = 4,00\%\end{aligned}$$

Contoh perhitungan pengujian dengan sampel data ke-6:

$$\begin{aligned}\text{Waktu tunda (ms)} &= \frac{\text{Sudut Penyalaan}}{180^\circ} \times 10 \\ &= \frac{135^\circ}{180^\circ} \times 10 = 7,50 \text{ ms}\end{aligned}$$

Nilai kesalahan yang didapat dari selisih *absolute* perhitungan dan pengujian sebesar:

$$\begin{aligned}\text{Kesalahan (ms)} &= |\text{Perhitungan} - \text{Pengujian}| \\ &= |7,50 - 7,00| = 0,50 \text{ ms}\end{aligned}$$

$$\begin{aligned}\text{Kesalahan (\%)} &= \frac{\text{Kesalahan}}{\text{Data Perhitungan}} \times 100\% \\ &= \frac{0,50}{7,50} \times 100\% = 6,67\%\end{aligned}$$

Hasil perhitungan kesalahan pengujian secara keseluruhan dapat dilihat dalam Tabel 5.7.

Tabel 5.7 Kesalahan Pengujian Sudut Penyalaan TRIAC

Sudut Penyalaan (°)	Perhitungan Waktu Tunda (ms)	Pengujian Waktu Tunda (ms)	Kesalahan (ms)	Kesalahan (%)
16	1,25	1,10	0,15	12,00
32	2,50	2,40	0,10	4,00
48	3,75	3,30	0,45	12,00
64	5,00	4,80	0,20	4,00
80	6,25	5,90	0,35	5,60
96	7,50	7,00	0,50	6,67
112	8,75	8,20	0,55	6,29
Rata-rata Kesalahan (%)				7,22

Dari hasil perhitungan kesalahan pengujian secara keseluruhan didapatkan nilai rata-rata kesalahan keseluruhan sebesar 7,22%. Kesalahan ini dikarenakan adanya proses perhitungan dan pembulatan angka oleh mikrokontroler dan sumber AC yang tidak ideal baik dari frekuensi dan bentuk gelombang tegangan, sehingga menyebabkan adanya selisih antara perhitungan

dan pengukuran. Selain itu proses pembacaan pada *oscilloscope* yang dilakukan secara manual juga dapat menimbulkan kesalahan pembacaan. Berdasarkan analisis diatas bisa disimpulkan bahwa mikrokontroler mampu menghasilkan sudut penyalaan sesuai dengan yang diberikan.

5.6 Pengujian Rangkaian *Dimmer AC*

5.6.1 Tujuan

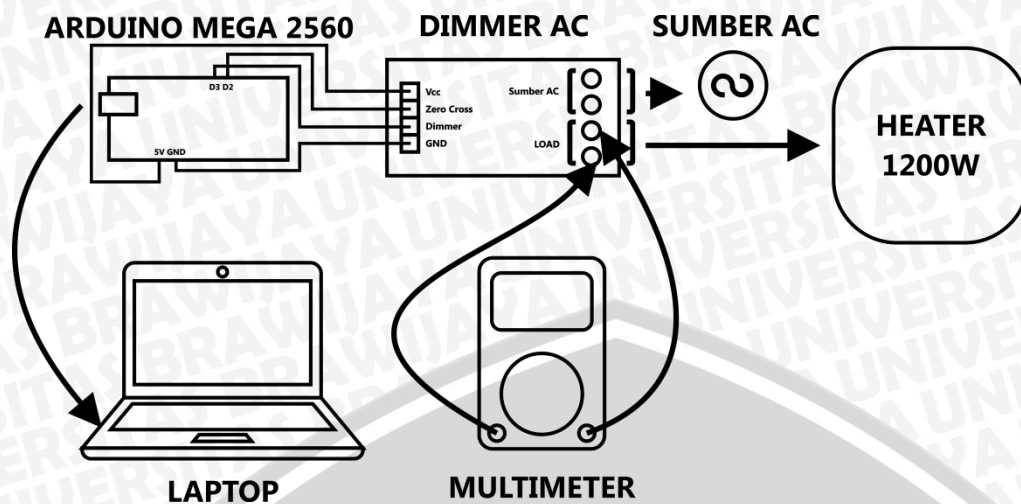
Pengujian ini dilakukan untuk mengetahui apakah rangkaian ini dapat digunakan sebagai *driver* untuk *heater* dengan membandingkan antara tegangan dengan perubahan dari PWM.

5.6.2 Peralatan yang digunakan

- Rangkaian *Dimmer AC*
- Arduino Mega 256
- Multimeter
- Sumber Tegangan AC
- *Heater* 1200 watt
- Laptop

5.6.3 Prosedur pengujian

1. Menyusun rangkaian seperti dalam Gambar 5.15.
2. Menghubungkan rangkaian *dimmer AC* dengan arduino.
3. Menghubungkan rangkaian *dimmer AC* dengan sumber AC.
4. Menghubungkan Arduino dengan laptop.
5. Memberikan beban pada rangkaian *dimmer AC*.
6. Memberikan nilai PWM pada rangkaian *dimmer AC* melalui *software* arduino ERW 1.6.5.
7. Mengukur tegangan yang dikeluarkan rangkaian *dimmer AC* kepada beban.



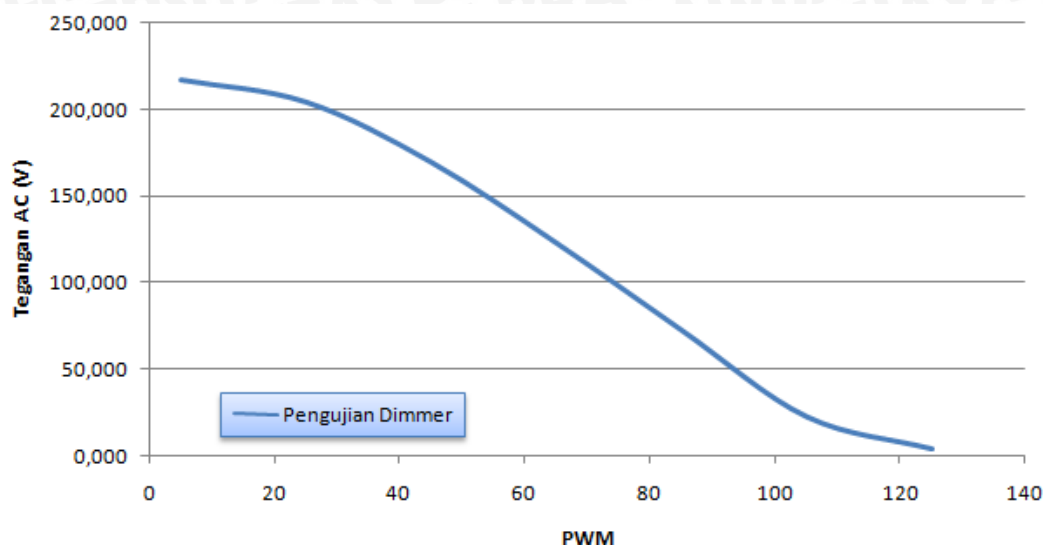
Gambar 5.15 Pengujian Rangkaian *Dimmer* AC (Perancangan)

5.6.4 Hasil pengujian

Dari hasil pengujian rangkaian *dimmer* AC didapatkan hasil berupa nilai tegangan yang berbeda-beda sesuai dengan nilai PWM yang diberikan. Hasil pengujian dapat dilihat dalam Tabel 5.8 dan Gambar 5.16.

Tabel 5.8 Perbandingan Nilai PWM Terhadap Tegangan (Perancangan)

Data ke-	PWM	Tegangan (VAC)
1	125	4,03
2	115	12,54
3	105	22,58
4	95	32,34
5	85	72,50
6	75	98,50
7	65	122,90
8	55	147,30
9	45	169,60
10	35	188,80
11	25	204,30
12	15	214,00
13	5	217,10



Gambar 5.16 Grafik Hubungan Nilai PWM dengan Besarnya Tegangan pada Elemen Pemanas (Perancangan)

Berdasarkan Tabel 5.8 dapat disimpulkan bahwa rangkaian *dimmer AC* dapat berfungsi dan dapat digunakan sebagai *driver* aktuator (elemen pemanas), ditandai dengan adanya perubahan tegangan akibat perubahan nilai PWM yang diberikan, sehingga bisa disimpulkan bahwa *dimmer AC* dapat digunakan sebagai *driver* untuk *heater* (elemen pemanas).

5.7 Pengujian Sistem Keseluruhan

5.7.1 Tujuan

Pengujian ini dilakukan untuk mengetahui kinerja *hardware* dan *software* serta untuk mengetahui respon keluaran sistem secara keseluruhan dengan menggunakan Algoritma Logika *Fuzzy* apakah bisa memberikan hasil respon keluaran seperti yang diinginkan.

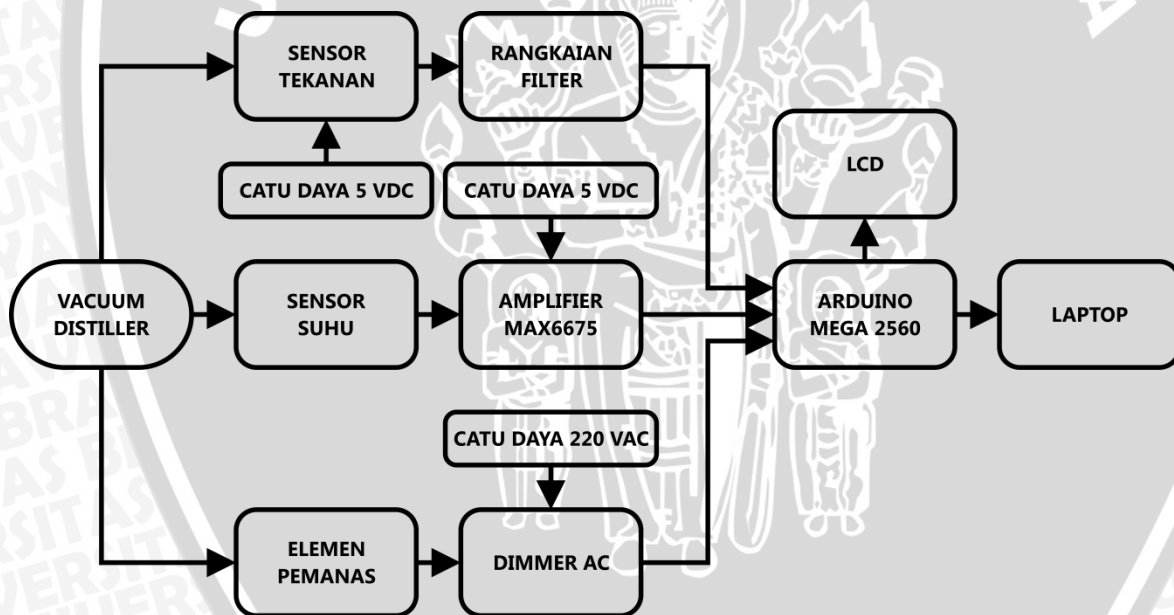
5.7.2 Peralatan yang digunakan

- *Vacuum Distiller*
- Elemen Pemanas
- Sensor suhu *K Type Thermocouple*
- Modul *Thermocouple Amplifier MAX6675*
- Sensor Tekanan MPX5100AP
- Rangkaian *Filter*

- Rangkaian *Dimmer* AC
- Sumber Tegangan AC
- Rangkaian LCD
- Arduino Mega 2560
- Laptop

5.7.3 Prosedur pengujian

1. Menghubungkan rangkaian seperti dalam Gambar 5.17.
2. Mengunduh program dengan Algoritma Logika *Fuzzy* lengkap dengan fungsi keanggotaan, *rule base*, dan metode *defuzzifikasi* yang telah didapat melalui *software* arduino ERW 1.6.5.
3. Menampilkan dan merekam nilai suhu dan tekanan dalam *vacuum distiller* pada *serial monitor software* arduino ERW 1.6.5.
4. Membuat grafik dari data yang didapat.

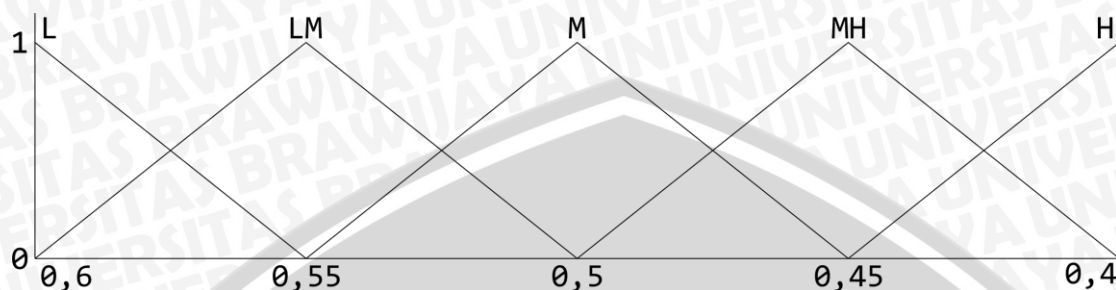


Gambar 5.17 Pengujian Rangkaian Keseluruhan (Perancangan)

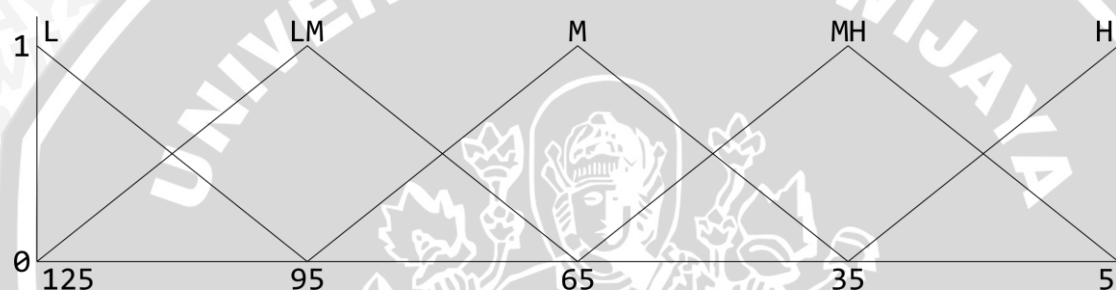
5.7.4 Hasil pengujian

Dari hasil pengujian keseluruhan dengan menggunakan 5 label fungsi keanggotaan masukan dan keluaran, *rule base* yang telah ditala ulang, dengan metode inferensi *Max-Min*, serta metode defuzzifikasi menggunakan *Weighted Average* (WA), didapatkan percobaan sebanyak empat kali dengan mengganti bentuk fungsi keanggotaan masukan *temperature error* yang difungsikan untuk

melihat respon keluaran terbaik sesuai dengan spesifikasi perancangan awal. Sedangkan fungsi keanggotaan masukan *vacuum pressure* yang ditunjukkan dalam Gambar 5.18 dan fungsi keanggotaan keluaran PWM yang ditunjukkan dalam Gambar 5.19 adalah tetap.

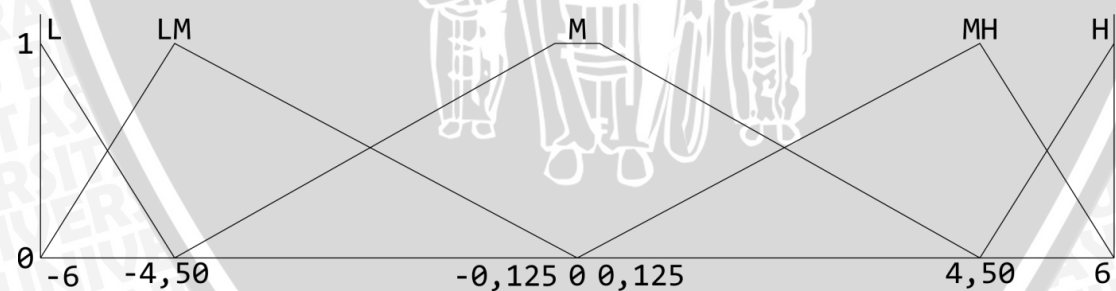


Gambar 5.18 Fungsi Keanggotaan Masukan *Vacuum Pressure* (Perancangan)

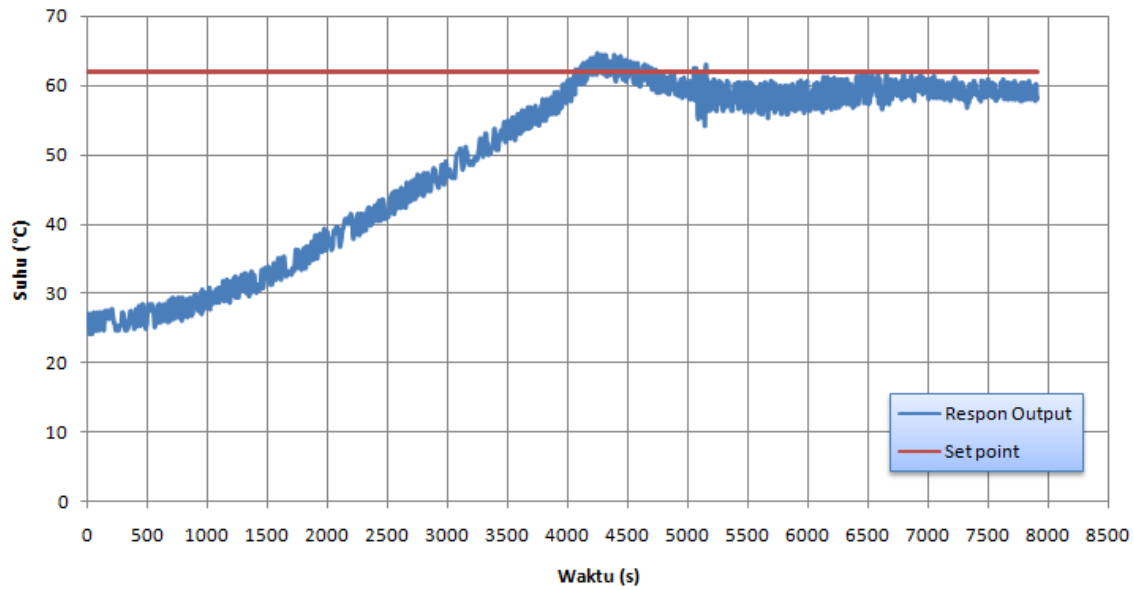


Gambar 5.19 Fungsi Keanggotaan Keluaran PWM (Perancangan)

Berikut merupakan hasil dari empat kali percobaan dari mengganti bentuk dari fungsi keanggotaan masukan *temperature error* serta *rule base* hasil penalaan ulang yang ditunjukkan dalam Tabel 5.9.

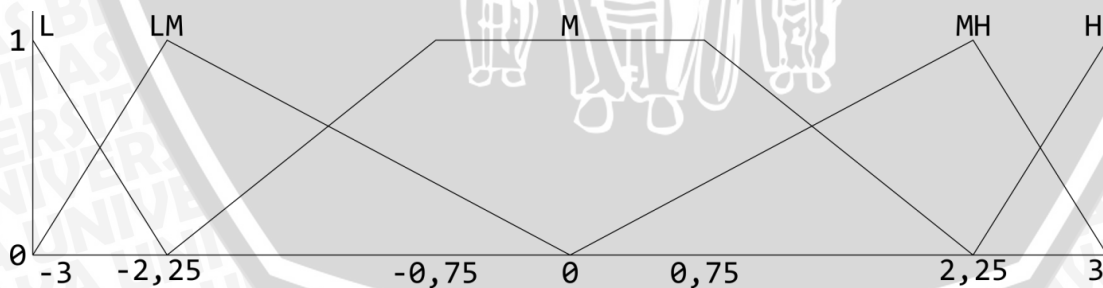


Gambar 5.20 Fungsi Keanggotaan Masukan *Temperature Error* Percobaan 1 (Perancangan)

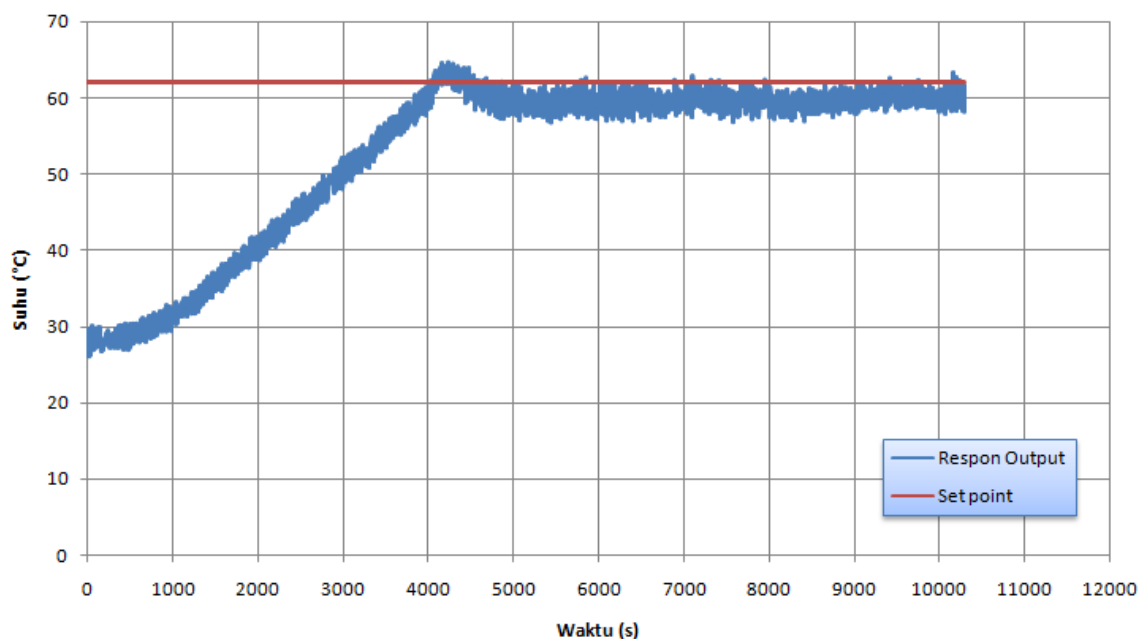


Gambar 5.21 Grafik Respon Keluaran Sistem Percobaan 1 (Perancangan)

Pada percobaan 1 ini dibentuk fungsi keanggotaan seperti yang ditunjukkan dalam Gambar 5.20. Dalam percobaan fungsi keanggotaan tersebut didapatkan hasil respon keluaran sistem yang ditunjukkan dalam Gambar 5.21. Respon keluaran sistem tersebut menunjukkan bahwa *error steady state* (e_{ss}) sangat besar yaitu 7,77%, *maximum overshoot* (M_p) juga besar yaitu 10,28%, dan juga *settling time* (t_s) yang cukup lama yaitu 6910 detik. Oleh sebab itu, dirancang kembali fungsi keanggotaan masukan *temperature error* untuk percobaan 2.

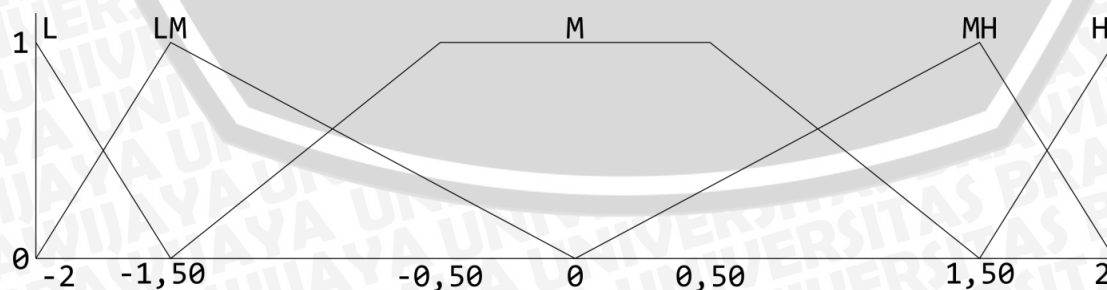


Gambar 5.22 Fungsi Keanggotaan Masukan *Temperature Error* Percobaan 2 (Perancangan)

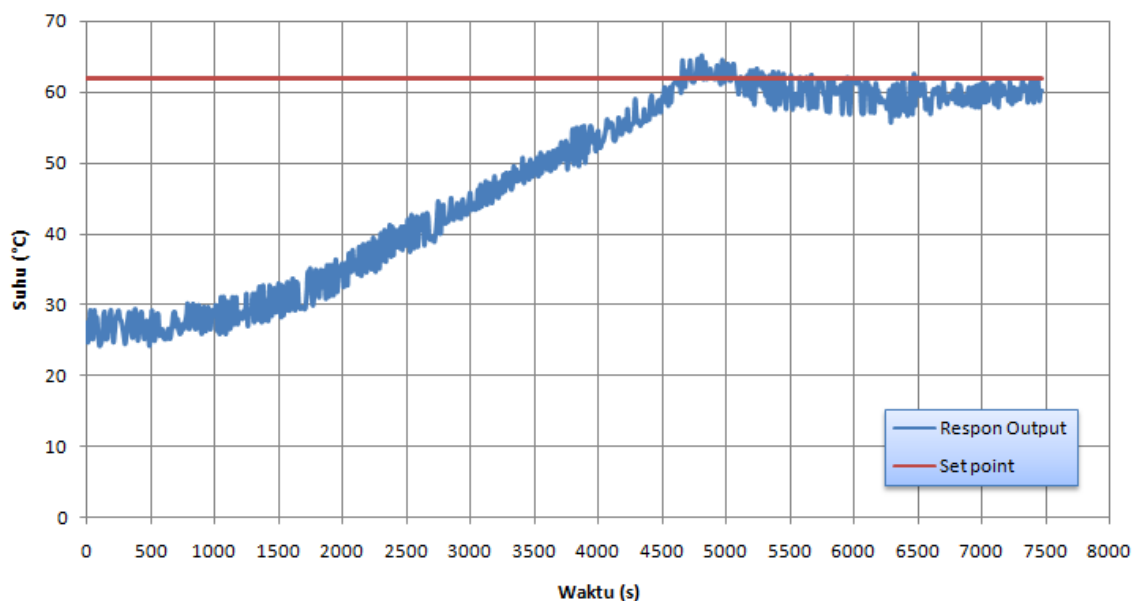


Gambar 5.23 Grafik Respon Keluaran Sistem Percobaan 2 (Perancangan)

Pada percobaan 2 ini dibentuk fungsi keanggotaan seperti yang ditunjukkan dalam Gambar 5.22. Dalam percobaan fungsi keanggotaan tersebut didapatkan hasil respon keluaran sistem yang ditunjukkan dalam Gambar 5.23. Respon keluaran sistem tersebut menunjukkan bahwa *error steady state* (e_{ss}) masih besar yaitu 6,29%, *maximum overshoot* (M_p) juga cukup besar yaitu 9,44%, dan juga *settling time* (t_s) yang lumayan cepat yaitu 4595 detik, tetapi hasil dari respon keluaran sistem lebih baik dibanding percobaan 1. Untuk memperoleh respon keluaran sistem yang lebih baik lagi maka dirancang kembali fungsi keanggotaan masukan *temperature error* untuk percobaan 3.

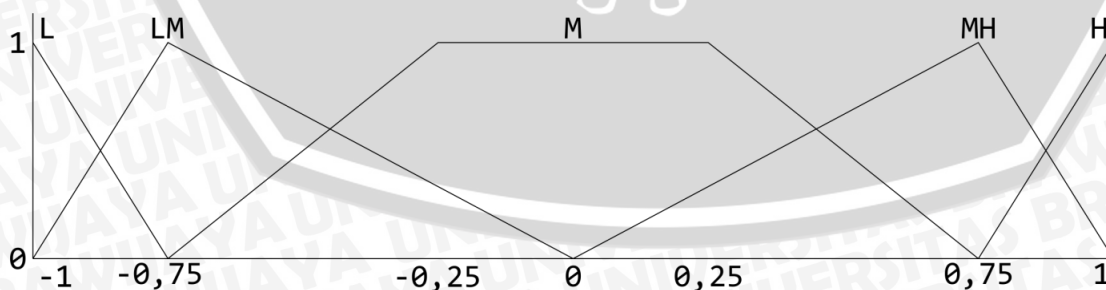


Gambar 5.24 Fungsi Keanggotaan Masukan *Temperature Error* Percobaan 3 (Perancangan)

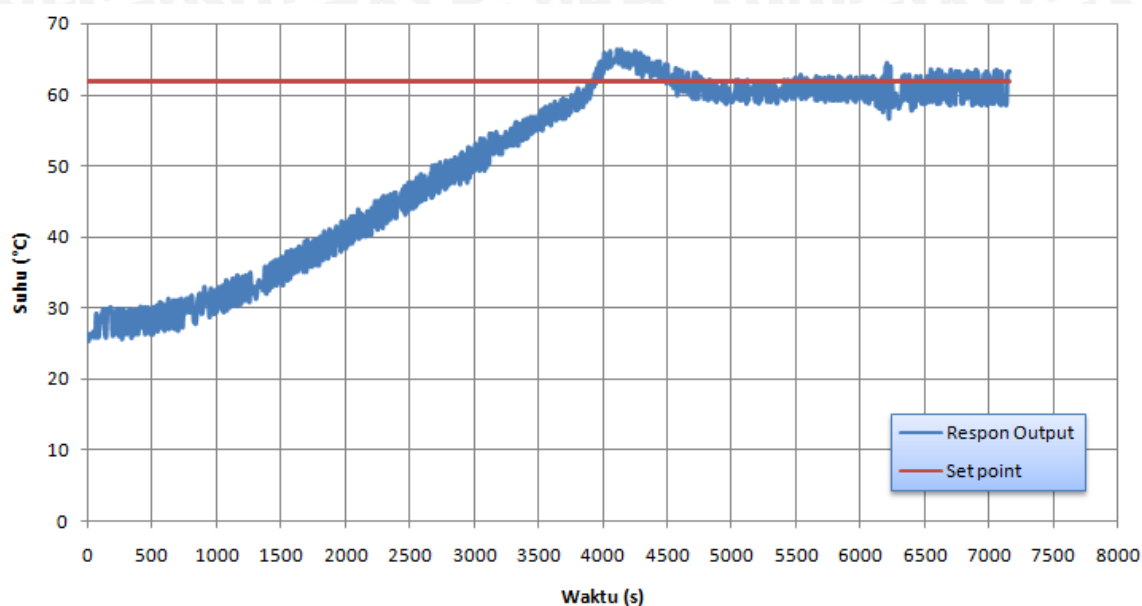


Gambar 5.25 Grafik Respon Keluaran Sistem Percobaan 3 (Perancangan)

Pada percobaan 3 ini dibentuk fungsi keanggotaan seperti yang ditunjukkan dalam Gambar 5.24. Dalam percobaan fungsi keanggotaan tersebut didapatkan hasil respon keluaran sistem yang ditunjukkan dalam Gambar 5.25. Respon keluaran sistem tersebut menunjukkan bahwa *error steady state* (e_{ss}) yang masih besar yaitu 6,08%, *maximum overshoot* (M_p) yang sudah mengecil yaitu 7,86%, dan juga *settling time* (t_s) yang lebih besar dari sebelumnya yaitu 6720 detik, tetapi hasil dari respon keluaran sistem lebih baik dibanding percobaan 2. Untuk memperoleh respon keluaran sistem yang lebih baik lagi maka dirancang kembali fungsi keanggotaan masukan *temperature error* untuk percobaan 3.



Gambar 5.26 Fungsi Keanggotaan Masukan *Temperature Error* Percobaan 4 (Perancangan)



Gambar 5.27 Grafik Respon Keluaran Sistem Percobaan 4 (Perancangan)

Pada percobaan 4 ini dibentuk fungsi keanggotaan seperti yang ditunjukkan dalam Gambar 5.26. Dalam percobaan fungsi keanggotaan tersebut didapatkan hasil respon keluaran sistem yang ditunjukkan dalam Gambar 5.27. Respon keluaran sistem tersebut menunjukkan bahwa *error steady state* (e_{ss}) sudah semakin mengecil dan sesuai dengan spesifikasi awal yaitu 3,85%, *maximum overshoot* (M_p) yang masih besar, tetapi sudah masuk dalam kriteria spesifikasi perancangan yaitu 9,85%, dan juga *settling time* (t_s) yang cukup cepat yaitu 4270 detik.

Tabel 5.9 Rule Base Fuzzy (Perancangan)

FUZZY RULE		Error				
		L	LM	M	MH	H
Vacuum Pressure	L	L	L	L	L	L
	LM	L	L	LM	M	M
	M	L	LM	M	M	MH
	MH	M	MH	MH	H	H
	H	H	H	H	H	H

Rangkuman dari empat kali percobaan tersebut dapat dilihat pada Tabel 5.10 berikut ini.

Tabel 5.10 Percobaan Respon Keluaran Sistem (Perancangan)

Percobaan	e_{ss} (%)	M_p (%)	t_s (s)	Keterangan
1	7,77	10,28	6910	Error = Range [-6 sampai 6]
2	6,29	9,44	4595	Error = Range [-3 sampai 3]
3	6,08	7,86	6720	Error = Range [-2 sampai 2]
4	3,85	9,85	4270	Error = Range [-1 sampai 1]

Berdasarkan hasil yang didapatkan dari Tabel 5.10, dapat disimpulkan bahwa percobaan ke-4 dengan *error steady state* = 3,85%, *maximum overshoot* = 9,85%, dan *settling time* = 4270 detik yang paling memenuhi dan sesuai dengan spesifikasi perancangan awal serta agar bisa menjaga suhu sama dengan *setpoint*.

Agar dapat mengetahui apakah respon keluaran sistem sesuai dengan tujuan (spesifikasi) perancangan awal, maka diperlukan perhitungan dengan sampel respon keluaran sistem percobaan ke-4 sebagai berikut:

a. *Error Steady State* (e_{ss})

Pada respon keluaran sistem percobaan ke-4 yang ditunjukkan dalam Gambar 5.27, saat keadaan *steady state* terjadi osilasi dengan nilai suhu tengah *steady state* ($T_{\text{tengah steady state}}$) 60,59°C dengan suhu mula-mula ($T_{\text{mula-mula}}$) adalah 25,41°C, serta suhu *setpoint* (T_{setpoint}) adalah 62°C, sehingga *error steady state* nya adalah:

$$\begin{aligned}
 e_{ss} (\%) &= \left| \frac{(T_{\text{tengah steady state}}) - (T_{\text{setpoint}})}{(T_{\text{setpoint}}) - (T_{\text{mula-mula}})} \right| \times 100\% \\
 &= \left| \frac{60,59 - 62}{62 - 25,41} \right| \times 100\% \\
 &= 3,85\%
 \end{aligned}$$

Keterangan:

- e_{ss} : *Error Steady State*
- $T_{\text{tengah steady state}}$: Suhu rata-rata dari waktu ke-4780 detik sampai dengan waktu ke-~ detik
- $T_{\text{mula-mula}}$: Suhu saat waktu pertama atau waktu ke-0 detik
- T_{setpoint} : Suhu yang diinginkan, yaitu 62°C

b. *Maximum Overshoot* (Mp)

Pada respon keluaran sistem percobaan ke-4 yang ditunjukkan dalam Gambar 5.27, terjadi *overshoot* hingga mencapai suhu tengah *overshoot* ($T_{\text{tengah overshoot}}$) 64,06°C dengan suhu mula-mula ($T_{\text{mula-mula}}$) adalah 25,41°C dan suhu tengah *steady state* ($T_{\text{tengah steady state}}$) adalah 60,59°C, sehingga perhitungan *maximum overshoot* nya adalah:

$$\begin{aligned} \text{Mp (\%)} &= \frac{(T_{\text{tengah overshoot}}) - (T_{\text{tengah steady state}})}{(T_{\text{tengah steady state}}) - (T_{\text{mula-mula}})} \times 100\% \\ &= \frac{64,06 - 60,59}{60,59 - 25,41} \times 100\% \\ &= 9,85\% \end{aligned}$$

Keterangan:

- Mp : *Maximum Overshoot*
- $T_{\text{tengah overshoot}}$: Suhu rata-rata dari waktu ke-3905 detik sampai dengan waktu ke-4180 detik
- $T_{\text{mula-mula}}$: Suhu saat waktu pertama atau waktu ke-0 detik
- $T_{\text{tengah steady state}}$: Suhu rata-rata dari waktu ke-4780 detik sampai dengan waktu ke-~ detik

c. *Settling Time* (t_s)

Pada respon keluaran sistem percobaan ke-4 yang ditunjukkan dalam Gambar 5.27, *settling time* terjadi pada detik ke-4270 atau 1,19 jam.

Berdasarkan analisis kinerja pengujian sistem secara keseluruhan, kesimpulan yang dapat diambil yaitu bahwa sistem pengontrolan suhu dengan struktur *Feedback-Feedforward* pada proses distilasi vakum bioetanol berbasis Algoritma Logika Fuzzy dapat berjalan dengan baik dan sesuai dengan tujuan serta spesifikasi perancangan awal.



BAB VI

PENUTUP

6.1 Kesimpulan

Penggunaan Algoritma Logika *Fuzzy* sebagai pengontrol suhu dengan struktur *Feedback-Feedforward* menggunakan 5 label fungsi keanggotaan untuk masukan *temperature error* dan *vacuum pressure*, serta 5 label fungsi keanggotaan keluaran PWM, dan juga menggunakan metode inferensi *Max-Min*, serta metode defuzzifikasi *Weighted Average* didapatkan *error steady state* sebesar 3,85% dan juga *maximum overshoot* sebesar 9,85%, serta *settling time* yang mencapai 4270 detik atau 1,19 jam. Dari hasil tersebut, maka bisa disimpulkan Algoritma Logika *Fuzzy* dapat menjaga suhu sama dengan *setpoint* serta telah sesuai dengan spesifikasi perancangan awal.

6.2 Saran

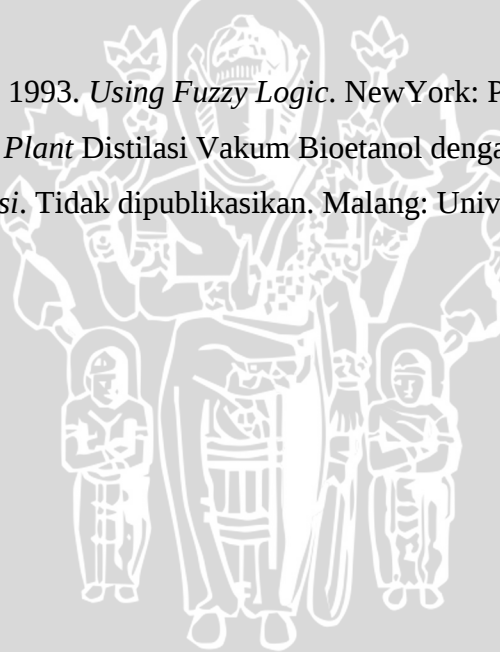
Pada penelitian selanjutnya disarankan agar mengganti media pemanas dari minyak goreng ke air, hal ini dikarenakan karakteristik dari minyak goreng apabila sudah panas (terlalu panas) akan sulit atau lama untuk dingin kembali (suhu turun), dan itu sangat berpengaruh terhadap suhu larutan di dalam tangki evaporator serta pada penentuan label dan juga bentuk dari fungsi keanggotaan nantinya. Selain itu, perlu ditala ulang *rule base* dengan kondisi yang berbeda serta ditambahkan label dari fungsi keanggotaan masukan dan keluaran agar semakin banyak kemungkinan kondisi yang tercipta dan semakin banyak pula aksi kontrol yang akan dipilih. Dan juga perlu adanya percobaan yang lebih banyak lagi agar bisa memperoleh respon keluaran sistem yang sesuai dengan spesifikasi yang diharapkan.



DAFTAR PUSTAKA

- Ackland, T. 2012. *Home Distillation of Alcohol*. Diakses dari <http://homedistiller.org> tanggal 17 Maret 2015.
- Ariassandi, R. 2014. Implementasi Kontrol Logika Fuzzy (KLF) sebagai Pengendali Suhu Sistem Pasteurisasi Kuning Telur Cair Berbasis Mikrokontroler Arduino. *Skripsi*. Tidak dipublikasikan. Malang: Universitas Brawijaya.
- Birol, F. 2006. *World Energy Prospects and Challenges*. Paper. Australian Economic Review, The University of Melbourne.
- Bolton, W. 2004. *Sistem Instrumentasi dan Sistem Kontrol (Judul asli: Instrumentation and Control System)*. Jakarta: Erlangga.
- Farkade, H. S., Pathre, A. P. 2012. "Experimental Investigation of metanol, etanol and Butanol Blends with Gasoline on SI Engine". *International Journal of Emerging Technology and Advanced Engineering*. 2(4): 205-215.
- Firsyari, A. S. 2014. Sistem Pengendalian Suhu pada Proses Distilasi Vakum Bioetanol dengan Menggunakan Arduino. *Skripsi*. Tidak dipublikasikan. Malang: Universitas Brawijaya.
- Inmojo. 2010. *Digital AC Dimmer Module*. <http://www.inmojo.com/store/inmojo-market/item/digital-ac-dimmer-module/>. (Diakses 7 Juli 2015).
- Muslim, M. A., Wahyudi, S., Nusantara, G. D., Abidin, Z. 2013. Pengembangan Rancang Bangun Sistem Pengendali Suhu dan Tekanan pada Proses Penyulingan Bioetanol sebagai Bahan Bakar Alternatif Kendaraan Bermotor. Malang: Universitas Brawijaya.
- Nixon, M., Mc. Gaw. 2001. *The Compleat Distiller*. Auckland: The Amphora Society
- Ogata, K.. 1985. *Teknik Kontrol Automatik (Sistem Pengaturan) Jilid 1 (Judul asli: Modern Control Engineering)*. Jakarta: Erlangga.
- Ogata, K. 1997. *Modern Control Engineering*. Prentice Hall. New Jersey.
- Philip, C. L. & Harbor, R. D. 1996. *Feedback Control System*. Diterjemahkan oleh Widodo.R.J. New Jersey: Prentice Hall.
- Primatama, A. 2013. Perancangan Soft Starter Motor Induksi Satu Fasa Dengan Metode Closed Loop Menggunakan Mikrokontroler Arduino.

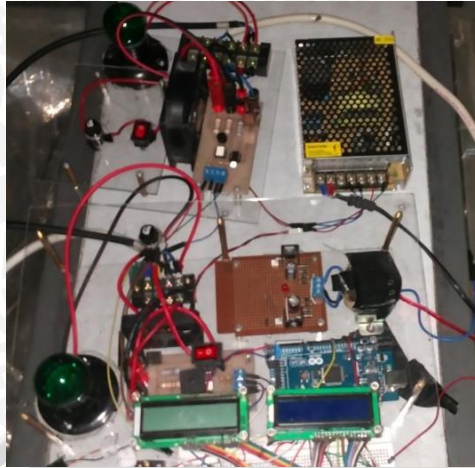
- Putri, T. W. O. 2014. Pengendalian Suhu pada Sistem Pasteurisasi Telur Cair Berbasis PLC (*Programmable Logic Controller*) Siemens Simatic S7-200 dan HMI (*Human Machine Interface*) Simatic HMI Panel. *Skripsi*. Tidak dipublikasikan. Malang: Universitas Brawijaya.
- Ramadhan, K. 2014. Sistem Kontrol Multivariabel Temperatur dan Level dengan Yokogawa DCS Centum VP. *Skripsi*. Tidak dipublikasikan. Malang: Universitas Brawijaya.
- Robert, T. 2013. Diakses dari <http://wiki.dxarts.washington.edu> tanggal 25 November 2013.
- Rusdi, A. P. Sistem Pengendalian Suhu Pada Proses Distilasi Vakum Bioetanol Menggunakan Kontrol Logika Fuzzy. *Skripsi*. Tidak dipublikasikan. Malang: Universitas Brawijaya.
- Wyman, C. E.. 1996. *Handbook on Bioethanol: Production and Utilization*. Washington, DC, Taylor & Francis.
- Yan, J., Ryan, M., dan Power, J. 1993. *Using Fuzzy Logic*. New York: Prentice Hall.
- Zainudin, M., 2015. Identifikasi *Plant* Distilasi Vakum Bioetanol dengan Metode *Extended Least Square* (ELS). *Skripsi*. Tidak dipublikasikan. Malang: Universitas Brawijaya.



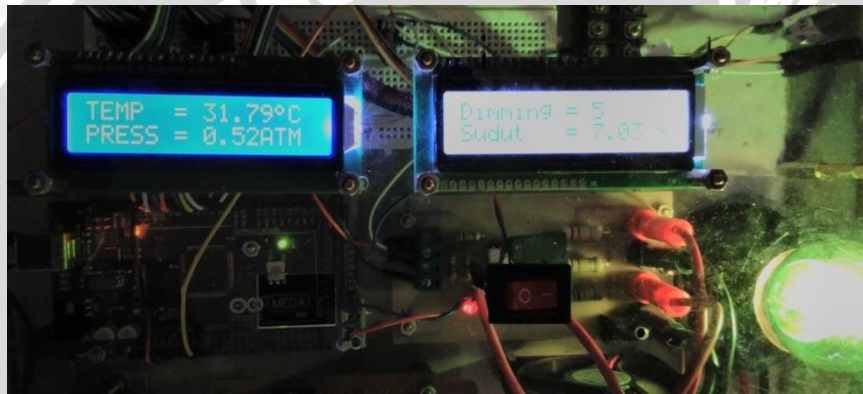
LAMPIRAN 1

FOTO ALAT





Gambar Rangkaian Elektris Proses Distilasi Vakum Bioetanol



Gambar Display Proses Distilasi Vakum Bioetanol



Gambar Alat Proses Distilasi Vakum Bioetanol

LAMPIRAN 2

LISTING PROGRAM



PROGRAM ARDUINO MEGA 2560

```

/*
 * LIBRARY
 */

#include <Wire.h>

// 1 :: MAX6675
#include <max6675.h>

// 1 :: KLF
#include<stdio.h>
#include<math.h>

#define L 0
#define LM 1
#define M 2
#define MH 3
#define H 4

// 1 :: LCD
#include <LiquidCrystal.h>

/*
 * DEKLARASI
 */

// 2 :: MAX6675
int thermo_gnd_pin = 45;
int thermo_vcc_pin = 47;
int thermo_so_pin = 49;
int thermo_cs_pin = 51;
int thermo_sck_pin = 53;

float v, w;
MAX6675 thermocouple(thermo_sck_pin, thermo_cs_pin, thermo_so_pin);
uint8_t degree[8] = {140,146,146,140,128,128,128,128};

// 2 :: MPX5100AP
const int PIN_MPX5100AP = A0;
int ADC_SEBELUM_KALIBRASI;
float PRESSURE_SEBELUM_KALIBRASI;
float PRESSURE_SETELAH_KALIBRASI;

// 2 :: KLF
int sp;

float terror;
float pvacuum;
float a;
float tError[6];
float pVacuum[6];
float Dimmer_fuzzy[25];

// 2 :: Dimmer
int AC_LOAD = 3;
int PWM;
float sudut;

// 2 :: Buat Objek LCD
LiquidCrystal lcd1(8,9,7,6,5,4);
LiquidCrystal lcd2(8,10,7,6,5,4);

/*

```

```

/* RULE BASE FUZZY
**/

unsigned char rule_Dimmer[5][5]=
{
    {L,L,L,L,L},
    {L,L,LM,M,M},
    {L,LM,M,M,MH},
    {M,MH,MH,H,H},
    {H,H,H,H,H},
};

float trapesium(int value, float x0, float x1, float x2, float x3);
float triangle(int value, float x0, float x1, float x2);
float reverse_grade(int value, float x0, float x1);
float grade(int value, float x0, float x1);
float fuzzyfikasi();
float rule_base_Dimmer();
float fuzzy_dimmer();

/*
 * SETUP
 */

void setup ()
{
    Serial.begin(9600);

    sp=62;

    // 3 :: MAX6675
    pinMode(thermo_vcc_pin, OUTPUT);
    pinMode(thermo_gnd_pin, OUTPUT);
    digitalWrite(thermo_vcc_pin, HIGH);
    digitalWrite(thermo_gnd_pin, LOW);

    lcd1.createChar(0, degree);

    // 3 :: Dimmer
    pinMode(AC_LOAD, OUTPUT);
    attachInterrupt(0, zero_crosss_int, RISING);

    // 3 :: LCD
    lcd1.begin(16,2);
    lcd2.begin(16,2);
}

/*
 * PROGRAM DIMMER
 */

void zero_crosss_int() // function to be fired at the zero crossing to
dim the light
{
    // Firing angle calculation :: 50Hz-> 10ms (1/2 cycle)
    // (10000us - 10us) / 128 = 75 (Approx)
    int dimtime = (75*PWM);
    delayMicroseconds(dimtime); // off cycle
    digitalWrite(AC_LOAD, HIGH); // triac firing
    delayMicroseconds(10); // triac on propogation delay
    digitalWrite(AC_LOAD, LOW); // triac off
}

/*
 * LOOPING
 */

void loop()
{
    // 4 :: MAX6675

```

```

v = thermocouple.readCelsius();
w = (-1.759384283)+(0.945179584*v);
terror=sp-w;

// 4 :: MPX5100AP
ADC_SEBELUM_KALIBRASI = analogRead(PIN_MPX5100AP);
PRESSURE_SEBELUM_KALIBRASI = ((ADC_SEBELUM_KALIBRASI*0.001273885)*1);
PRESSURE_SETELAH_KALIBRASI =
(0.137413074+(0.818845619*PRESSURE_SEBELUM_KALIBRASI)); pvacuum =
PRESSURE_SETELAH_KALIBRASI;

// 4 :: KLF
fuzzyfikasi();

// 4 :: Dimmer
PWM = (int)fuzzy_dimmer();

// LCD :: MAX6675
Serial.print("tBefore:");
Serial.print("\t");
Serial.print(v);
Serial.print("\t");
Serial.print("tAfter:");
Serial.print("\t");
Serial.print(w);
Serial.print("\t");

lcd1.clear();
lcd1.setCursor(0, 0);
lcd1.print("TEMP = ");

lcd1.print(w);
#if ARDUINO >= 100
lcd1.write((byte)0);
#else
lcd1.print(0,BYTE);
#endif
lcd1.print("C");

lcd1.setCursor(0,1);
lcd1.print("PRESS = ");
lcd1.print(pvacuum);
lcd1.print("ATM");

// LCD :: MPX5100AP
Serial.print("pADC:");
Serial.print("\t");
Serial.print(ADC_SEBELUM_KALIBRASI);
Serial.print("\t");
Serial.print("pBefore:");
Serial.print("\t");
Serial.print(PRESSURE_SEBELUM_KALIBRASI);
Serial.print("\t");
Serial.print("pAfter:");
Serial.print("\t");
Serial.print(pvacuum);
Serial.print("\t");

// LCD :: Dimmer
Serial.print("PWM:");
Serial.print("\t");
Serial.print(PWM);
Serial.print("\t");
Serial.print("Sudut:");
Serial.print("\t");
Serial.print(sudut);
Serial.print("\n");

sudut=(PWM*1.40625);
lcd2.clear();

```

```

lcd2.print("PWM = ");
lcd2.print(PWM);
lcd2.setCursor(0,1);
lcd2.print("Sudut = ");
lcd2.print(sudut);
lcd2.print(" *");

delay(1000);
delay(1000);
delay(1000);
delay(1000);
delay(1000);
}

/*
 *PROGRAM FUZZY
 */

float fuzzyfikasi()
{
    tError[L]=reverse_grade(terror,-1,-0.75);
    tError[LM]=triangle(terror,-1,-0.75,0);
    tError[M]=trapesium(terror,-0.75,-0.25,0.25,0.75);
    tError[MH]=triangle(terror,0,0.75,1);
    tError[H]=grade(terror,0.75,1);

    pVacuum[L]=reverse_grade(pvacuum,0.6,0.55);
    pVacuum[LM]=triangle(pvacuum,0.6,0.55,0.5);
    pVacuum[M]=triangle(pvacuum,0.55,0.5,0.45);
    pVacuum[MH]=triangle(pvacuum,0.5,0.45,0.4);
    pVacuum[H]=grade(pvacuum,0.45,0.4);
}

float fuzzy_dimmer()
{
    fuzzyfikasi();
    return rule_base_Dimmer();
}

float rule_base_Dimmer()
{
    float umin;
    float umax;
    float Num,Denum;
    float num[]={0,0,0,0,0};
    float denum[]={0,0,0,0,0};
    int x,y;
    float centre;
    float nilai_Dimmer;
    float Dimmer_out=0;

    for(x=L;x<=H;x++)
    {
        for(y=L;y<=H;y++)
        {
            if(tError[x]>0 && pVacuum[y]>0)
            {
                umin=(float)fmin(tError[x],pvacuum[y]);
                Dimmer_fuzzy[rule_Dimmer[x][y]]=(float)fmax(umin,Dimmer_fuzzy[rule_Dimmer[x][y]]);

                if (rule_Dimmer[x][y]==0)
                {
                    centre=125;
                }
                else if(rule_Dimmer[x][y]==1)
                {
                    centre=95;
                }
                else if(rule_Dimmer[x][y]==2)
            }
        }
    }
}

```

```

        {
            centre=65;
        }
        else if(rule_Dimmer[x][y]==3)
        {
            centre=35;
        }
        else
        {
            centre=5;
        }

        num[rule_Dimmer[x][y]]=Dimmer_fuzzy[rule_Dimmer[x][y]]*centre;
        denum[rule_Dimmer[x][y]]=Dimmer_fuzzy[rule_Dimmer[x][y]];
    }
}

Num = num[0]+num[1]+num[2]+num[3]+num[4];
Denum = denum[0]+denum[1]+denum[2]+denum[3]+denum[4];
Dimmer_out=Num/Denum;
return Dimmer_out;
}

```

```

/*
 * FUNGSI KEANGGOTAAN FUZZY
 */

```

```

float trapesium(int value, float x0, float x1, float x2, float x3)
{
    float result=0;
    float x;
    x=value;

    if ((x<=x0)|| (x>=x3))
    {
        result=0;
    }

    else if(x>x1 && x<x2)
    {
        result=1;
    }

    else if((x>=x0)&&(x<x1))
    {
        result=((x-x0)/(x1-x0));
    }

    else
    {
        result=(((x-x3)/(x3-x2)));
    }

    return result;
}

```

```

float triangle(int value, float x0, float x1, float x2)
{
    float result=0;
    float x;
    x=value;

    if ((x<=x0)|| (x>=x2))
    {
        result=0;
    }

    else if(x==x1)

```



```
{
    result=1;
}
else if((x>=x0)&&(x<x1))
{
    result=((x-x0)/(x1-x0));
}
else
{
    result=(((x-x0)+x2)/(x2-x1));
}
return result;
}
float grade(int value, float x0, float x1)
{
    float result=0;
    float x;
    x=value;
    if(x<=x0)
    {
        result=0;
    }
    else if(x>=x1)
    {
        result=1;
    }
    else
    {
        result=(x-x0)/(x1-x0);
    }
    return result;
}
float reverse_grade(int value, float x0, float x1)
{
    float result =0;
    float x;
    x=value;
    if(x<=x0)
    {
        result=1;
    }
    else if(x>=x1)
    {
        result=0;
    }
    else
    {
        result=(-x+x1)/(x1-x0);
    }
    return result;
}
```

