

PENGENDALIAN SUHU PADA PROSESOR LAPTOP

**MENGGUNAKAN KONTROL LOGIKA FUZZY BERBASIS
MIKROKONTROLER ARDUINO MEGA**

SKRIPSI

JURUSAN TEKNIK ELEKTRO

Diajukan untuk memenuhi persyaratan
memperoleh Gelar Sarjana Teknik



DISUSUN OLEH :

FAUZAN NUSYURA

NIM. 105060300111070 - 63

KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN

UNIVERSITAS BRAWIJAYA

FAKULTAS TEKNIK

JURUSAN TEKNIK ELEKTRO

MALANG

2015

LEMBAR PERSETUJUAN

PENGENDALIAN SUHU PADA PROSESOR LAPTOP
MENGGUNAKAN KONTROL LOGIKA FUZZY BERBASIS
MIKROKONTROLER ARDUINO MEGA

SKRIPSI

JURUSAN TEKNIK ELEKTRO

Diajukan untuk memenuhi persyaratan
memperoleh Gelar Sarjana Teknik



DISUSUN OLEH :

FAUZAN NUSYURA

NIM. 105060300111070 - 63

Telah diperiksa dan disetujui oleh :

Pembimbing 1

Pembimbing 2

Ir. Retnowati, MT.
NIP. 19511224 198203 2 001

Dr. Ir. Erni Yudaningtyas, MT.
NIP. 19650913 199002 2 001

LEMBAR PENGESAHAN

PENGENDALIAN SUHU PADA PROSESOR LAPTOP MENGGUNAKAN KONTROL LOGIKA FUZZY BERBASIS MIKROKONTROLER ARDUINO MEGA

SKRIPSI JURUSAN TEKNIK ELEKTRO

Disusun oleh:

FAUZAN NUSYURA
NIM. 105060300111070 – 63

Skrripsi ini telah diuji dan dinyatakan lulus pada
tanggal 13 Januari 2015

MAJELIS PENGUJI

Goegoes Dwi Nusantoro, S.T., M.T.
NIP. 19711013 200604 1 001

Ir. Purwanto, M.T.
NIP. 19540424 198601 1 001

Rahmadwati, ST., MT., Ph.D
NIP. 19771102 200604 2 003

Mengetahui
Ketua Jurusan Teknik Elektro

M. Aziz Muslim, S.T., M.T., Ph.D.
NIP. 19741203 200012 1 001

PENGANTAR

Alhamdulillah, puji syukur penulis panjatkan kehadirat Allah SWT yang telah memberikan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi ini. Tak lepas shalawat serta salam tercurahkan kepada junjungan kita Nabi Muhammad SAW selaku suri tauladan bagi yang mengharapkan rahmat dan hidayah-Nya.

Penyusunan skripsi ini tidak lepas dari bantuan, bimbingan serta dorongan dari berbagai pihak. Pada kesempatan yang baik ini penulis ingin menyampaikan rasa terima kasih kepada:

- Orang tua penulis, Bapak Wiyono dan Ibu Anik Susilaningsih yang telah merawat, memberikan kasih sayang, membesarakan, membimbing, dan mendoakan kelancaran penulis hingga terselesaikannya skripsi ini.
- Adik penulis Fauzi Ilham Akbar yang selalu saling membantu dan memberi semangat kepada penulis.
- Bapak M. Aziz Muslim, ST., MT., Ph.D selaku Ketua Jurusan Teknik Elektro Universitas Brawijaya dan Dosen Pembimbing Akademik penulis atas segala nasehat dan bimbingan yang telah diberikan.
- Bapak Hadi Suyono, ST., MT., Ph.D selaku Sekretaris Jurusan Teknik Elektro Universitas Brawijaya.
- Bapak Ir. Purwanto, MT. selaku KKDK Teknik Kontrol Jurusan Teknik Elektro UB atas segala ilmu, bimbingan, nasehat, gagasan, ide, saran, motivasi, dan bantuan yang telah diberikan.
- Ibu Ir. Retnowati, MT., selaku Dosen Pembimbing 1 atas segala ilmu, bimbingan, nasehat, gagasan, ide, saran, motivasi, dan bantuan yang telah diberikan.
- Ibu Dr. Ir. Erni Yudaningtyas, MT., selaku Dosen Pembimbing 2 atas segala ilmu, bimbingan, nasehat, gagasan, ide, saran, motivasi, dan bantuan yang telah diberikan.
- Bapak Ibu Dosen, karyawan, staf recording dan RBTE atas segala bantuan dan kemudahan.



- Sahabat Dwelspada dan sahabat SMAN 2 Malang Pret, Bagus, Bekti, Bobi, Candra, Yogi, Nuri, Leli, Indri, Desti dkk yang selama ini juga selalu menghibur dan memberi support kepada penulis.
- Sahabat HMH : Deo, Basori, Kibul, Embik, Samto, Comel, Abu, Ares, Ari, Azis, Boni, Dwi Yoga, Faishol, Fajar, Fery, Gosi, Imam, Kacong, Kevin, Maman, Sesa, Samid, Yatma, Yoga dan seterusnya yang tidak bisa saya sebutkan satu-satu atas bantuan semangatnya.
- Sahabat RisTIE : Sam Averus, Sam Lalu, Sam Pram, Sam Sean, Amiril, Bayu, Deo, Dayat, Lele , Tempei, Nizar, Rohman, Reno, Bagus, Lalu, Priyo, Rahman, Randy, Riza, Ridha, Risto, Else dkk atas semangat dan bantuannya.
- Rekan di Elektro : Reza, Agam, Ahlan, Arez, Firman, Inal, Ponari, Lukman, Dewi, Erni, Nuni, Reta dkk yang membantu selama perkuliahan dan organisasi di Elektro.
- Sahabat seperjuangan skripsi yang selalu membantu dan mendukung skripsi penulis, Samid, Lele, Boni, Isol, Sugab, Nawaw, Abutempe dan teman teman lain yang juga saling membantu untuk mencapai gelar Sarjana Teknik.
- Seluruh Keluarga Besar Magnet'10, yang selalu memberi pengalaman-pengalaman, saran-saran dan dukungan-dukungan yang membangun selama perkuliahan dan penggerjaan skripsi.
- Seluruh Keluarga Besar Jurusan Teknik Elektro Universitas Brawijaya, serta semua pihak yang tidak mungkin untuk dicantumkan namanya satu per satu, terima kasih banyak atas bantuan dan dukungannya.

Penulis menyadari bahwa skripsi ini belum sempurna karena keterbatasan ilmu dan kendala yang terjadi selama penggerjaan skripsi. Oleh karena itu, penulis terbuka terhadap kritik dan saran untuk penyempurnaan tulisan di masa yang akan datang. Penulis juga berharap tulisan ini dapat bermanfaat bagi kita semua.

Malang, Januari 2015

Penulis

ABSTRAK

Fauzan Nusyura, Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya, Januari 2015, Pengendalian Suhu pada Prosesor Laptop Menggunakan Kontrol Logika *Fuzzy* Berbasis Mikrokontroler Arduino Mega, Dosen Pembimbing: Ir. Retnowati, MT., Dr. Ir. Erni Yudaningtyas, MT.

Semakin banyak daya yang digunakan laptop maka baterai yang digunakan akan cepat terkuras. Salah satu contohnya adalah penggunaan daya yang banyak pada kipas internal. Pada penelitian ini berfokus untuk membantu dan mengurangi kerja kipas internal laptop. Agar daya yang digunakan baterai semakin berkurang. Penelitian ini menggunakan sensor LM35 sebagai sensor utama untuk parameter pengendali suhu pada laptop. Arduino Mega digunakan sebagai alat pengontrol utama dengan Kontrol Logika *Fuzzy* (KLF).

Proses perancangan KLF pada penelitian ini menggunakan *5 Membership Function* (MF) dengan metode Inferensi *MIN-MAX Composition* dan metode Defuzzifikasi *Center of Gravity* (COG). Pada proses pengendalian, didapatkan hasil respon sistem yang sesuai dengan *settling time* sebesar 196 detik. Respon pada kondisi *steady state* menghasilkan suhu 48,78 °C. Proses pengujian pada sistem mendapat *error steady state* sebesar 1,47 %.

Waktu pemakaian aktual baterai mulai dari kapasitas 98 % s/d 20 %, adalah 149 menit atau 2 jam 29 menit. Sistem mampu mempertahankan baterai laptop lebih lama sebesar 48 menit atau 2880 detik dan baterai berhemat sebesar 47,52 % dengan menggunakan alat ini.

Kata kunci : Laptop, Prosesor, Pengendalian Suhu, Kontrol Logika *Fuzzy*, Arduino Mega.



DAFTAR ISI

PENGANTAR	iv
ABSTRAK	vi
DAFTAR ISI.....	vii
DAFTAR GAMBAR.....	ix
DAFTAR TABEL	x
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	2
1.5 Sistematika Penulisan.....	3
BAB II TINJAUAN PUSTAKA.....	4
2.1 Laptop.....	4
2.2 Prosesor	5
2.2.1 Fungsi Prosesor	5
2.2.2 Temperatur Prosesor	6
2.3 Motor DC	7
2.4 Logika Fuzzy	8
2.4.1 Struktur Dasar Kontrol Logika Fuzzy	9
2.4.2 Fungsi Keanggotaan.....	9
2.4.3 Kontroler Logika Fuzzy.....	11
2.5 Arduino Mega 2560.....	14
2.5.1 Tegangan Sumber.....	14
2.5.2 Memori.....	14
2.5.3 Input dan Output	15
2.5.4 Komunikasi	15
2.6 Sensor Suhu LM35	16
2.7 <i>Driver</i> Motor DC	16
2.8 Sinyal PWM	17
2.9 Liquid Crystal Display (LCD) Keypad Shield	18
BAB III METODE PENELITIAN	19
3.1 Perancangan Sistem.....	19



3.2	Pembuatan Sistem	19
3.3	Pengujian dan Analisa Data	20
3.4	Pengambilan Kesimpulan.....	20
BAB IV	PERANCANGAN DAN PEMBUATAN ALAT	21
4.1	Diagram Blok Sistem	21
4.2	Spesifikasi Alat.....	22
4.3	Prinsip Kerja Sistem.....	22
4.4	Perancangan Perangkat Keras	23
4.4.1	Kotak Pengendali Suhu	23
4.4.2	Rangkaian <i>Liquid Crystal Display (LCD)</i>	23
4.4.3	Rangkaian Driver Motor	24
4.4.4	Rangkaian Mikrokontroler	25
4.5	Perancangan Perangkat Lunak	25
4.5.1	<i>Flowchart</i> Sistem Secara Keseluruhan	25
4.5.2	<i>Flowchart</i> Pembacaan Sensor suhu	26
4.5.3	<i>Flowchart</i> Kontroler Logika <i>Fuzzy</i>	27
4.5.4	<i>Flowchart</i> Putaran Motor.....	30
4.5.5	<i>Flowchart</i> Tampilan <i>Liquid Crystal Display (LCD)</i>	31
4.5.6	Perancangan Kontroler Logika <i>Fuzzy</i>	32
BAB V	PENGUJIAN DAN ANALISIS.....	35
5.1	Pengujian Penentuan Setpoint	35
5.2	Pengujian LCD	36
5.3	Pengujian Sensor LM35	37
5.4	Pengujian Driver Motor DC	39
5.5	Pengujian Laptop Tanpa Sistem Pengendali	41
5.6	Pengujian Sistem Secara Keseluruhan	44
BAB VI	PENUTUP	48
6.1	Kesimpulan.....	48
6.2	Saran	48
DAFTAR PUSTAKA		49
LAMPIRAN.....		50



DAFTAR GAMBAR

Gambar 2.1 Laptop Acer Aspire 8920	5
Gambar 2.2 (a) Pin mikroprosesor Intel 80486DX2, (b) Diagram blok sederhana sebuah CPU	6
Gambar 2.3 Monitoring suhu prosesor dengan menggunakan Program Speccy versi 1.26.698	7
Gambar 2.4 Cara Kerja Motor DC	8
Gambar 2.5 Pengendali Fuzzy.....	9
Gambar 2.6 Fungsi Keanggotaan Bentuk Triangular	10
Gambar 2.7 Fungsi Keanggotaan Bentuk Trapesium.....	11
Gambar 2.8 Inferensi Fuzzy dengan Metode MAX-MIN	13
Gambar 2.9 Arduino mega tampak atas	14
Gambar 2.10 Skema LM35	16
Gambar 2.11 IC <i>driver</i> motor L298N.....	17
Gambar 2.12 Sinyal PWM secara umum	17
Gambar 2.13 LCD Keypad Shield.....	18
Gambar 4.1 Blok diagram sistem	21
Gambar 4.2 Skema perancangan sistem pengendalian suhu pada prosesor	22
Gambar 4.3 Rangkaian <i>Liquid Crystal Display</i> (LCD)	24
Gambar 4.4 <i>Flowchart</i> keseluruhan sistem	26
Gambar 4.5 <i>Flowchart</i> pembacaan sensor suhu	27
Gambar 4.6 <i>Flowchart</i> kontroler logika <i>fuzzy</i>	28
Gambar 4.7 <i>Flowchart</i> subrutin <i>fuzzy</i>	28
Gambar 4.8 <i>Flowchart</i> subrutin <i>check rule</i>	29
Gambar 4.9 <i>Flowchart</i> subrutin <i>defuzzy</i>	30
Gambar 4.10 <i>Flowchart</i> putaran motor	31
Gambar 4.11 <i>Flowchart</i> Tampilan <i>Liquid Crystal Display</i> (LCD)	32
Gambar 4.12 (a) Fungsi keanggotaan <i>error</i> (b) Fungsi keanggotaan <i>delta error</i>	33
Gambar 4.13 Fungsi keanggotaan keluaran	33
Gambar 5.1 Grafik Hubungan Suhu dengan RPM kipas internal	36
Gambar 5.2 Tampilan Data Karakter pada LCD	37
Gambar 5.3 Grafik Perbandingan Suhu Prosesot dengan Suhu LM35	39
Gambar 5.4 Grafik Hubungan PWM terhadap Sudut Motor Servo	41
Gambar 5.5 Pengujian Laptop tanpa Sistem Pengendali.....	42
Gambar 5.6 Pemakaian Awal Baterai Laptop tanpa Sistem Pengendali	43
Gambar 5.7 Pemakaian Akhir Baterai Laptop tanpa Sistem Pengendali	44
Gambar 5.8 Pengujian Sistem Secara Keseluruhan.....	45
Gambar 5.9 Pemakaian Awal Baterai Laptop dengan Sistem Secara Keseluruhan	46
Gambar 5.10 Pemakaian Akhir Baterai Laptop dengan Sistem Secara Keseluruhan	47



DAFTAR TABEL

Tabel 4.1 Konfigurasi pin <i>driver</i> motor pada Arduino Mega	25
Tabel 4.2 Konfigurasi pin Arduino Uno.....	25
Tabel 4.3 Aturan <i>Fuzzy</i> pada Sistem	34
Tabel 5.1 Hasil Pengujian Suhu Prosesor Laptop	35
Tabel 5.2 Hasil Pengujian Sensor LM35 Sebelum Dikalibrasi	38
Tabel 5.3 Hasil Pengujian Sensor LM35 Setelah Dikalibrasi	39
Tabel 5.4 Hasil Pengujian <i>Driver</i> Motor L298N.....	41
Tabel 5.5 Data hasil pengujian sistem	54



BAB I

PENDAHULUAN

1.1 Latar Belakang

Di era teknologi yang semakin canggih ini tidak lepas dari yang namanya komputer jinjing atau biasa disebut laptop. Seperti namanya, laptop dapat dibawa kemana-mana dengan gampang, karena bentuknya yang relatif kecil dibandingkan dengan komputer pribadi atau *Personal Computer* (PC). Komputer berguna untuk banyak hal, contohnya untuk mengerjakan tugas, untuk bekerja, untuk menjelajah internet, untuk bermain, dan masih banyak lagi.

Di dalam komputer terdapat banyak sekali komponen elektronik yang saling terhubung satu sama lain. Salah satu komponen yang paling penting dalam komputer adalah *Central Processing Unit* (CPU) atau lebih dikenal dengan nama prosesor. Dengan tidak adanya CPU, maka komputer tidak akan bekerja atau menyala.

Temperatur pada prosesor bergantung pada proses yang dijalankan. Semakin banyak proses yang dijalankan prosesor, maka temperaturnya juga akan semakin tinggi, begitu juga sebaliknya. Karena ruang sirkulasi udara laptop sangat kecil, maka laptop lebih cepat panas dibandingkan PC.

Dengan tingginya temperatur dari prosesor, maka sistem akan memproses kipas dalam komputer untuk berputar lebih cepat agar udara panas yang berada dalam komputer dapat terbuang. Namun dengan berputarnya kipas menjadi lebih cepat, maka sistem akan mengambil daya yang lebih besar. Sebenarnya saat ini sudah tercipta kipas ekternal laptop yang biasa disebut dengan *Cooling Pad*, yang berguna untuk menurunkan temperatur prosesor. Namun dengan sumber tegangan 5 Volt yang diambil dari laptop, pastinya akan mengambil daya yang lebih besar lagi.

Sehubungan dengan masalah di atas, maka pada penulis ingin memberikan suatu inovasi dengan merancang alat yang dapat mengendalikan suhu prosesor pada laptop. Sistem pengendalian suhu pada penelitian ini menggunakan Mikrokontroler Arduino. Kontrol Logika Fuzzy (KLF) diterapkan pada Mikrokontroler Arduino sebagai pengendali sistem ini. Karena KLF memiliki beberapa keunggulan seperti sistemnya fleksibel dan toleransi terhadap data yang tidak akurat, maka dengan terciptanya alat ini, diharapkan suhu prosesor laptop akan berada pada suhu yang stabil. Sehingga pemakaian laptop dengan menggunakan sumber tegangan baterai dapat diperpanjang.



1.2 Rumusan Masalah

Mengacu pada permasalahan yang disebutkan pada latar belakang, maka ditetapkan rumusan masalah sebagai berikut :

1. Bagaimana merancang dan membuat sistem pengendalian suhu pada laptop menggunakan KLF ?
2. Bagaimana unjuk kerja baterai laptop, jika menggunakan sistem pengendalian suhu pada laptop menggunakan KLF ?

1.3 Batasan Masalah

Karena luasnya objek pengkajian sehingga perlu adanya pembatasan masalah agar pembahasan lebih terfokus pada rumusan masalah. Adapun batasan masalah pada skripsi ini adalah :

1. Perancangan dan pembuatan alat pengendalian suhu pada prosesor laptop yang digunakan adalah desain sendiri.
2. Menggunakan 3 kipas angin motor DC 12V sebagai aktuator.
3. Pembahasan ditekankan pada pengendalian suhu pada sistem menggunakan KLF berbasis Mikrokontroler Arduino Mega.
4. Temperatur yang diukur berkisar 25-75 °C dengan menggunakan sensor suhu LM35.
5. Kinerja *driver* dan rangkaian elektrik tidak dibahas mendalam.
6. Frekuensi yang digunakan sebesar 931,1 MHz dari Prosesor Intel Core i3-350M @ 2.26 GHz.
7. Laptop yang digunakan adalah Merk ASUS A42JC.

1.4 Tujuan

Tujuan dari penelitian ini adalah dapat terciptanya alat pengendali suhu pada prosesor laptop, sehingga alat tersebut dapat digunakan untuk memperpanjang waktu pemakaian laptop dengan menggunakan baterai.

1.5 Sistematika Penulisan

Adapun sistematika dari penulisan skripsi ini adalah:

BAB I Pendahuluan

Menjelaskan tentang latar belakang, rumusan masalah, batasan masalah, tujuan, dan sistematika penulisan.

BAB II Tinjauan Pustaka

Menjelaskan dasar teori penunjang penelitian yang ada pada alat ini, yang terdiri atas laptop, prosesor, motor DC, logika *fuzzy*, Arduino Mega, sensor suhu, driver motor DC, Sinyal PWM dan LCD.

BAB III Metode Penelitian

Membahas metode penelitian dan perencanaan alat.

BAB IV Perancangan Sistem

Membahas perancangan alat yang meliputi spesifikasi, perencanaan blok diagram, prinsip kerja, dan pembuatan alat. Setelah itu, bagaimana penerapannya dalam sistem secara keseluruhan.

BAB V Pengujian Alat

Membahas hasil pengujian sistem dan analisis data secara keseluruhan terhadap alat yang telah direalisasikan.

BAB VI Kesimpulan dan Saran

Menjelaskan tentang pengambilan kesimpulan sesuai dengan hasil perancangan dan pengujian alat serta saran-saran yang diperlukan untuk melakukan pengembangan aplikasi selanjutnya.



BAB II TINJAUAN PUSTAKA

Untuk memudahkan dalam memahami cara kerja rangkaian maupun dasar-dasar perencanaan alat ini, maka perlu adanya penjelasan dan uraian teori penunjang yang digunakan dalam penulisan ini. Teori penunjang yang akan dijelaskan dalam bab ini adalah:

- Laptop
- Prosesor
- Motor DC
- Logika Fuzzy
- Arduino Mega 2560
- Sensor Suhu LM35
- Driver Motor DC
- Sinyal PWM
- *Liquid Crystal Display (LCD) Keypad Shield*

2.1 Laptop

Laptop atau komputer jinjing adalah komputer bergerak yang berukuran relatif kecil dan ringan, beratnya berkisar dari 1-6 kg, bergantung kepada ukuran, bahan, dan spesifikasi laptop tersebut. Sumber daya laptop berasal dari baterai atau adaptor A/C yang dapat digunakan untuk mengisi ulang baterai dan menyalakan laptop itu sendiri. Baterai laptop pada umumnya dapat bertahan sekitar 1 hingga 6 jam sebelum akhirnya habis, tergantung dari cara pemakaian, spesifikasi, dan ukuran baterai. Laptop terkadang disebut juga dengan komputer notebook atau notebook saja.

Sebagai komputer pribadi, laptop memiliki fungsi yang sama dengan komputer destop / PC pada umumnya. Komponen yang terdapat di dalamnya sama persis dengan komponen pada destop, hanya saja ukurannya diperkecil, dijadikan lebih ringan, lebih tidak panas, dan lebih hemat daya.

Komputer jinjing kebanyakan menggunakan layar *Liquid Crystal Display (LCD)* berukuran 10 inci hingga 17 inci tergantung dari ukuran laptop itu sendiri. Selain itu, papan ketik yang terdapat pada laptop juga kadang-kadang dilengkapi dengan papan sentuh yang berfungsi sebagai pengganti *mouse*. Papan ketik dan *mouse* tambahan dapat dipasang melalui soket Universal Serial Bus maupun PS/2 jika tersedia.



Berbeda dengan komputer desktop, laptop memiliki komponen pendukung yang didesain secara khusus untuk mengakomodasi sifat komputer jinjing yang portabel. Sifat utama yang dimiliki oleh komponen penyusun laptop adalah ukuran yang kecil, hemat konsumsi energi, dan efisien. Komputer jinjing biasanya berharga lebih mahal, tergantung dari merek dan spesifikasi komponen penyusunnya, walaupun demikian harga komputer jinjing pun semakin mendekati desktop seiring dengan semakin tingginya tingkat permintaan konsumen. Gambar 2.1 merupakan contoh laptop dengan merek Acer dan versinya adalah Apire 8920 yang sedang menyala.



Gambar 2.1 Laptop Acer Aspire 8920 (id.wikipedia.org)

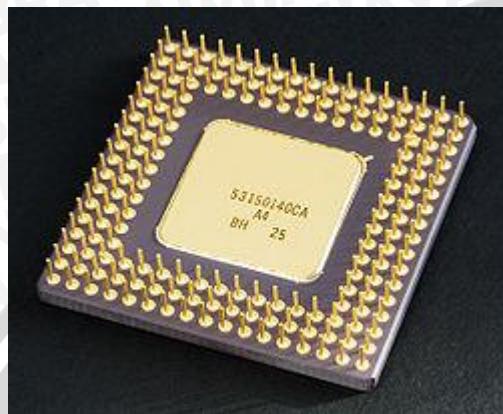
2.2 Prosesor

Central Processing Unit (CPU) atau biasa disebut *Central Processor Unit* adalah perangkat keras dalam komputer yang memahami dan melaksanakan perintah yang dikirim oleh sistem komputer. Pada umumnya, jenis CPU Laptop dapat diketahui dengan cara melihat pada bagian bawah keyboard.

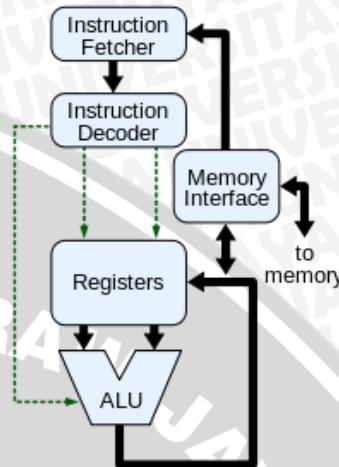
2.2.1 Fungsi Prosesor

Fungsi utama dari prosesor adalah melakukan operasi aritmatika dan logika terhadap data yang diambil dari memori atau dari informasi yang dimasukkan melalui beberapa perangkat keras, seperti papan tombol, pemindai, tuas kontrol, maupun dari mouse. Prosesor dikontrol menggunakan sekumpulan instruksi perangkat lunak komputer. Perangkat lunak tersebut dapat dijalankan oleh prosesor dengan membacanya dari media penyimpan, seperti cakram keras, disket, cakram padat, maupun pita perekam. Instruksi-instruksi tersebut kemudian disimpan

terlebih dahulu pada memori fisik (MAA), yang mana setiap instruksi akan diberi alamat unik yang disebut alamat memori. Selanjutnya, prosesor dapat mengakses data-data pada MAA dengan menentukan alamat data yang diinginkan. Gambar 2.2 merupakan contoh dari mikroprosesor dan diagram bloknya.



(a)



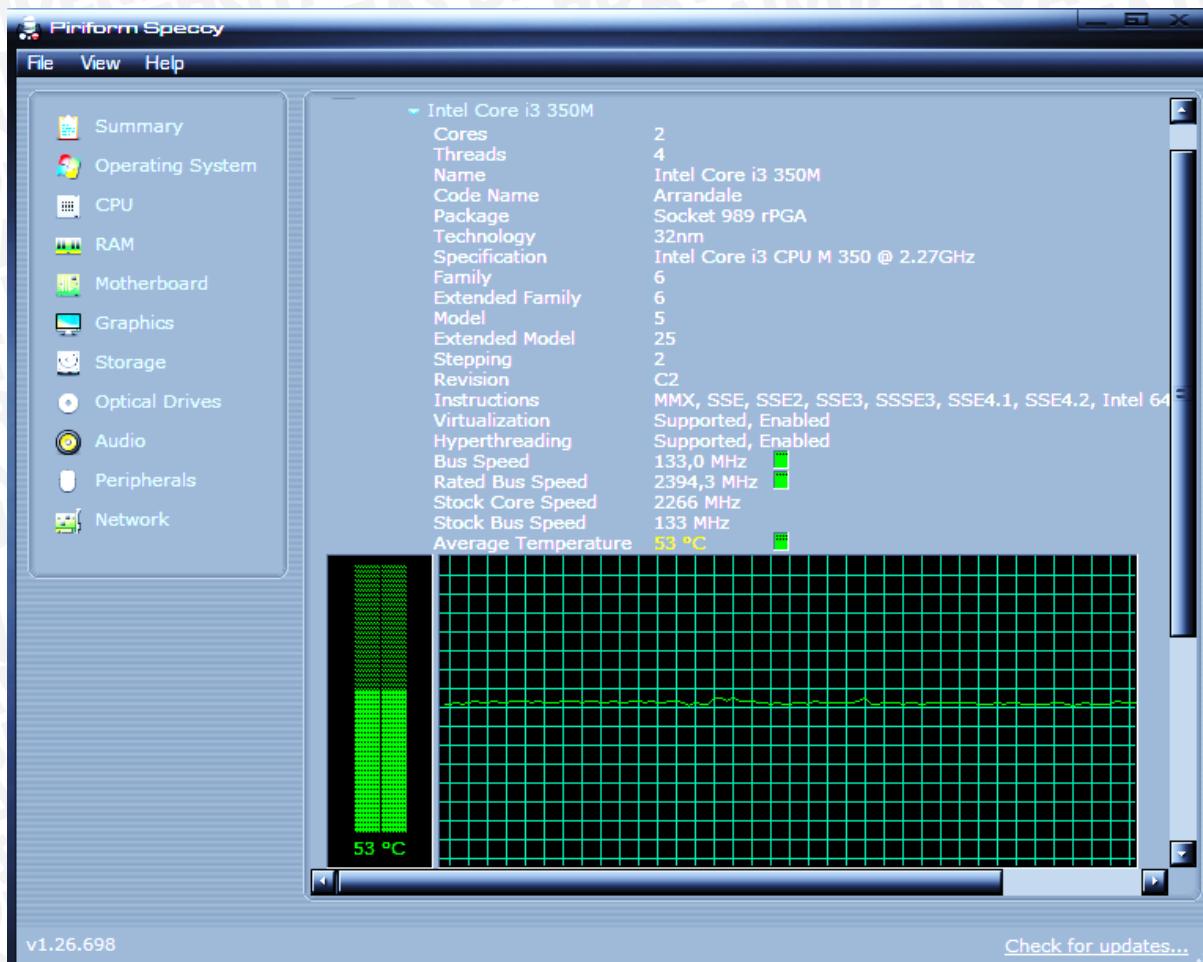
(b)

Gambar 2.2 (a) Pin mikroprosesor Intel 80486DX2, (b) Diagram blok sederhana sebuah CPU
(id.wikipedia.org)

2.2.2 Temperatur Prosesor

Saat ini telah banyak versi prosesor. Yang membedakan adalah dari seberapa cepat proses yang dapat dilakukan oleh prosesor tersebut. Semakin cepat kecepatan pemrosesan sebuah prosesor akan mempengaruhi suhu pada prosesor tersebut. Prosesor yang akan dikaji nanti adalah prosesor jenis Intel Core i3-350M yang memiliki kecepatan 2.26 Ghz. Saat prosesor menggunakan kecepatan prosesor sebesar 931,1 MHz, prosesor tersebut memiliki suhu kerja normal mulai dari 53 °C – 56 °C saat laptop dalam keadaan beroperasi normal (tanpa *Cooling Pad*).

Untuk lebih jelasnya dapat dilihat dalam Gambar 2.3 dibawah. Dapat dilihat bahwa suhu prosesor saat ini berada pada suhu sebesar 53 °C. Dari grafik dalam Gambar 2.3 dapat dilihat bahwa suhu rata-rata prosesor saat ini berkisar antara 53 °C – 56 °C.



Gambar 2.3 Monitoring suhu prosesor dengan menggunakan Program Speccy versi 1.26.698
(Laptop milik penulis)

2.3 Motor DC

Prinsip kerja motor DC sesuai dengan hukum kemagnetan Lorentz, yaitu membangkitkan fungsi magnet pada suatu konduktor berarus dalam medan magnet sehingga timbul ggl induksi. Setiap arus yang mengalir melalui sebuah konduktor akan menimbulkan medan magnet. Arah medan magnet dapat ditentukan dengan kaidah tangan kiri.

Kaidah tangan kiri untuk motor menunjukkan arah arus yang mengalir didalam sebuah konduktor yang berada dalam medan magnet. Jari tengah menunjukkan arah arus yang mengalir pada konduktor, jari telunjuk menunjukkan arah medan magnet dan ibu jari menunjukkan arah medan putar. Adapun besarnya gaya yang bekerja pada konduktor tersebut dapat dirumuskan dengan :

$$F = B \cdot I \cdot L \text{ (Newton)}$$

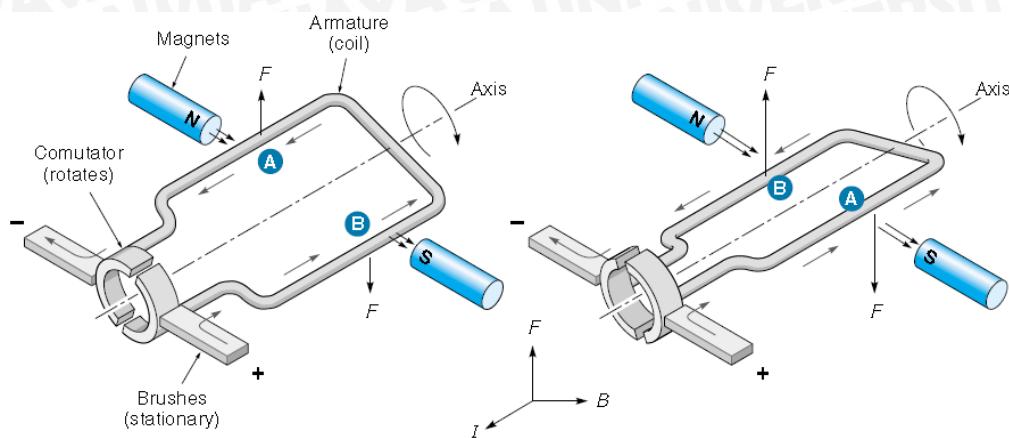
Dimana :

B = kerapatan fluks magnet (weber)

L = panjang konduktor (meter)

$I =$ arus listrik (ampere)
(Soemarwanto, 1999)

Gambar 2.4 adalah ilustrasi cara kerja motor DC yang mempunyai satu lilit kawat A-B berada di dalam medan magnet. Lilitan ini dapat berputar dengan bebas, lilitan ini biasa disebut dengan jangkar (armature).



Gambar 2.4 Cara Kerja Motor DC (Kilian, 1996)

Pada jangkar diberikan arus yang berasal dari sumber yang terhubung dengan sikat (*brushes*). Sikat-sikat ini terpasang pada sebuah cincin yang terbelah dua, yang disebut cincin belah (*commutator*). Adapun tujuan dari konstruksi ini adalah agar lilitan kawat dapat berputar apabila ada arus listrik yang melewatkinya.

Pada kawat yang berada di kanan arus mengalir dari depan ke belakang. Pada kawat yang berada di bagian kiri, arus mengalir dari belakang ke depan kawat a dan b secara bergantian berada di kiri dan kanan. Karena itu arah arus di a dan arah arus di b selalu bersifat bolak-balik. Pembalikan arah arus itu terjadi pada saat lilitan kawat melintasi posisi vertikal.

Bagian *commutator* berfungsi sebagai penyearah mekanik. Fluksi magnet yang ditimbulkan magnet permanen disebut medan magnet motor. Dalam Gambar 2.4 arah fluks magnetik adalah dari kiri ke kanan. Adapun gaya yang bekerja pada penghantar b adalah ke atas, sementara gaya yang bekerja pada penghantar a adalah ke bawah. Gaya-gaya yang bekerjasama kuatnya, sehingga terdapat kopel yang bekerja pada kawat sehingga lilitan jangkar dapat berputar. Setelah berputar 180° arah arus berbalik, pada saat itu penghantar a dan b bertukar tempat. Akibatnya arah gerak putaran tidak berubah.

2.4 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

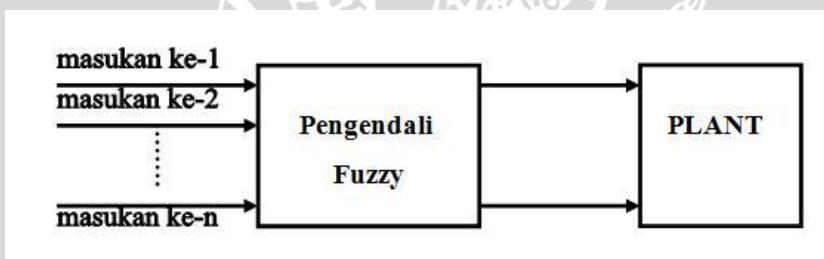
akrab daripada nilai tegas. Temperatur tertentu biasa dinyatakan sebagai panas, agak panas, atau sangat dingin daripada dinyatakan dalam nilai terukur tertentu.

Tahun 1965 L.A. Zadeh memodifikasi teori himpunan yang disebut himpunan kabur (*fuzzy Set*). Himpunan *fuzzy* di dasarkan 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 1982 pengendalian 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 (Sivanandam, 2006).

2.4.1 Struktur Dasar Kontrol Logika *Fuzzy*

Dalam sistem pengendalian dengan logika *fuzzy* dilibatkan suatu blok pengendali yang menerima satu atau lebih masukan dan mengumpulkan satu atau lebih keluaran ke plant atau blok lain sebagaimana ditunjukkan dalam Gambar 2.5.



Gambar 2.5 Pengendali *Fuzzy* (Coughanowr,1991)

Komponen utama penyusun Kontrol Logika *Fuzzy* adalah unit fuzzifikasi, *fuzzy inference*, dan unit defuzzifikasi. Basis pengetahuan terdiri dari dua jenis (Yan, 1994).

2.4.2 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 $[0,1]$. Secara matematika hal ini dinyatakan dengan :

$$\mu_A : U \rightarrow [0,1]$$

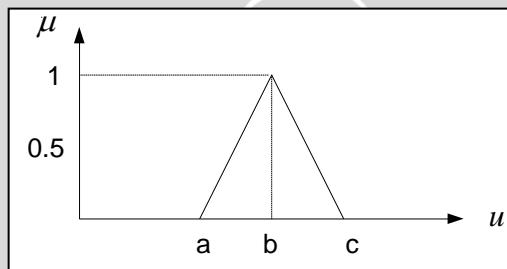
(Kuswadi, 2000)

➤ Fungsi keanggotaan bentuk Triangular

Definisi fungsi triangular sebagai berikut:

$$T(u; a, b, c) = \begin{cases} 0 & u < a \\ \frac{u-a}{b-a} & a \leq u \leq b \\ \frac{c-u}{c-b} & b \leq u \leq c \\ 0 & u > c \end{cases} \quad (2.4.1)$$

Fungsi keanggotaan bentuk Tringular ditunjukkan dalam Gambar 2.6



Gambar 2.6 Fungsi Keanggotaan Bentuk Triangular (Yan, 1994)

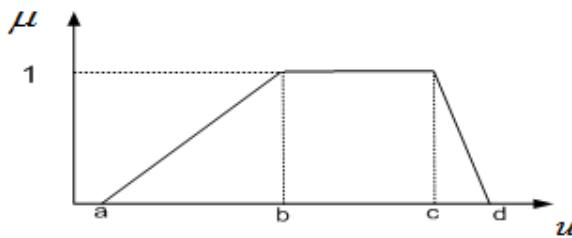
Fungsi keanggotaan bentuk triangular ini digunakan bila diinginkan himpunan fuzzy mempunyai nilai proporsional terhadap nol maupun satu.

➤ Fungsi keanggotaan bentuk Trapesium

Definisi fungsi trapesium sebagai berikut :

$$T(u; a, b, c, d) = \begin{cases} 0 & u < a \\ \frac{u-a}{b-a} & a \leq u \leq b \\ 1 & b \leq u \leq c \\ \frac{d-u}{d-c} & c \leq u \leq d \\ 0 & d \leq u \end{cases} \quad (2.4.2)$$

Fungsi keanggotaan bentuk Trapesium ditunjukkan pada Gambar 2.7.



Gambar 2.7 Fungsi Keanggotaan Bentuk Trapesium (Yan, 1994)

2.4.3 Kontroler Logika Fuzzy

Kontroler 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*.

Kontroler 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*.

2.4.3.1 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 pengendali 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.

2.4.3.2 Kaidah Aturan Fuzzy (Fuzzy Rule)

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

Dalam pengendali berbasis *fuzzy*, aturan pengendalian *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 A₁ AND Y is B₁ THEN Z is C₁

Rule 2 IF X is A₂ AND Y is B₂ THEN Z is C₂

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).

2.4.3.3 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*. Persamaan aturan minimum adalah

$$\mu_{C'} = \bigcup_1^n \alpha_i \wedge \mu_{ci} \quad (2.4.3)$$

dengan $\alpha_i = \mu_{Ai}(x_0) \wedge \mu_{Bi}(y_0)$

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

R_1 : Jika x adalah A_1 dan y adalah B_1 maka z adalah C_1

R_2 : Jika x adalah A_2 dan y adalah B_2 maka z adalah C_2

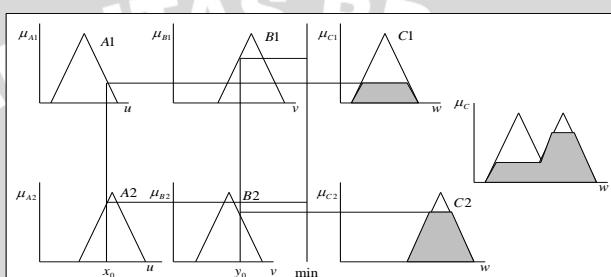
Pada metode penalaran MAX-MIN fungsi keanggotaan konsekuensi dinyatakan dengan

$$\mu_{C'}(W) = \mu_{c'1} \vee \mu_{c'2} = [\alpha_1 \wedge \mu_{c1}(w)] \vee [\alpha_2 \wedge \mu_{c2}(w)] \quad (2.4.4)$$

$$\text{Dimana } \alpha_1 = \mu_{A1}(x_0) \wedge \mu_{B1}(y_0) \quad (2.4.5)$$

$$\alpha_2 = \mu_{A2}(x_0) \wedge \mu_{B2}(y_0) \quad (2.4.6)$$

Lebih jelas metode ini dideskripsikan dalam Gambar 2.8.



Gambar 2.8 Inferensi Fuzzy dengan Metode MAX-MIN (Yan, 1994)

2.4.3.4 Metode Defuzzifikasi *Center Of Gravity* (COG)

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

$$y_0 = \text{defuzzifier}(y) \quad (2.4.7)$$

dengan:

y : aksi kontrol *fuzzy*

y_0 : aksi kontrol *crisp*

defuzzifier : operator defuzzifikasi

Metode COG didefinisikan sebagai berikut:

$$U = \frac{\sum_{i=1}^n w_i u_i}{\sum_{i=1}^n w_i} \quad (2.4.8)$$



dengan:

U = Keluaran

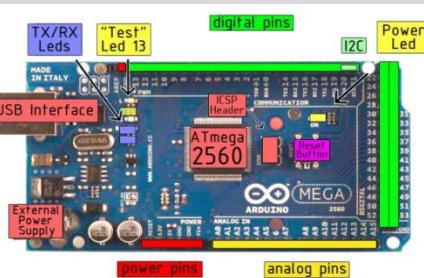
w_i = Bobot nilai benar w_i

u_i = Nilai linguistik pada fungsi keanggotaan keluaran

n = Banyak derajat keanggotaan

2.5 Arduino Mega 2560

Arduino Mega 2560 (dapat dilihat dalam Gambar 2.9) adalah papan mikrokontroler berdasarkan ATmega328. Board ini memiliki 54 digital *input/output* pin (14 pin dapat digunakan sebagai *output* PWM), 16 *input* analog, 16 MHz osilator kristal, USB koneksi, jack listrik, header ICSP, dan tombol reset.



Gambar 2.9 Arduino mega tampak atas (Datasheet Arduino Mega 2560)

2.5.1 Tegangan Sumber

Arduino mega 2560 dapat diaktifkan melalui koneksi USB atau dengan catu daya eksternal (otomatis). Eksternal (non-USB) daya dapat berasal baik dari AC ke adaptor DC atau baterai. Adaptor ini dapat dihubungkan dengan menancapkan *plug jack* pusat-positif ukuran 2.1 mm konektor power. Ujung kepala dari baterai dapat dimasukkan kedalam Gnd dan Vin pin *header* dari konektor power. Arduino dapat beroperasi dengan catu daya eksternal 6 sampai 20 volt. Namun jika menggunakan lebih dari 12V, regulator tegangan bisa panas dan merusak papan. Kisaran yang disarankan adalah 7 sampai 12 volt.

2.5.2 Memori

ATmega 2560 memiliki 256 KB (dengan 8 KB digunakan untuk *bootloader*), 8 KB dari SRAM dan 4 KB EEPROM.

2.5.3 Input dan Output

Masing-masing dari 54 pin digital di Arduino mega 2560 dapat digunakan sebagai *input* atau *output*, dengan menggunakan fungsi *pinMode ()*, *digitalWrite ()*, dan *digitalRead ()*, beroperasi dengan daya 5 volt. Setiap pin dapat memberikan atau menerima maksimum 40 mA dan memiliki internal *pull-up* resistor (secara default terputus) dari 20-50 Kohm. Selain itu, beberapa pin memiliki fungsi khusus:

- *Serial*: 0 (RX) dan 1 (TX); *Serial 1*: 19 (RX) dan 18 (TX); *Serial 2*: 17 (RX) dan 16 (TX); *Serial 3*: 15 (RX) dan 14 (TX). Digunakan untuk menerima (RX) dan mengirimkan (TX) TTL data serial. Pin ini dihubungkan ke pin yang berkaitan dengan chip Serial ATmega8U2 USB-to-TTL.
- *Eksternal interrupts*: 2 (*interrupt 0*), 3 (*interrupt 1*), 18 (*interrupt 5*), 19 (*interrupt 4*), 20 (*interrupt 3*), dan 2 (*interrupt 2*). Pin ini dapat dikonfigurasi untuk memicu *interrupt* pada nilai yang rendah, dengan batasan tepi naik atau turun, atau perubahan nilai.
- *PWM*: 0 - 13. Menyediakan output PWM 8-bit dengan fungsi *analogWrite ()*.
- *SPI*: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). Pin ini mendukung komunikasi SPI menggunakan *SPI library*.
- *LED*: 13. Ada *built-in* LED terhubung ke pin digital 13. Ketika pin bernilai nilai HIGH, LED menyala dan ketika pin bernilai LOW, LED mati.
- *I²C*: 20 (SDA) dan 21 (SCL). Dukungan I²C (TWI) komunikasi menggunakan *Wire*.

2.5.4 Komunikasi

Arduino Uno memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, arduino lain, atau mikrokontroler lainnya. Arduino mega 2560 menyediakan 4 UART TTL (5V) untuk komunikasi serial. Sebuah arduino mega 2560 sebagai saluran komunikasi serial melalui USB dan sebagai *port virtual com* untuk perangkat lunak pada komputer. *Firmware* '8 U2 menggunakan driver USB standar COM, dan tidak ada *driver* eksternal yang diperlukan. Namun, pada Windows diperlukan, sebuah file inf. Perangkat lunak Arduino terdapat monitor serial yang memungkinkan digunakan memonitor data tekstual sederhana yang akan dikirim ke atau dari board Arduino. LED RX dan TX di papan tulis akan

berkedip ketika data sedang dikirim melalui chip USB-to-serial dengan koneksi USB ke komputer (tetapi tidak untuk komunikasi serial pada pin 0 dan 1).

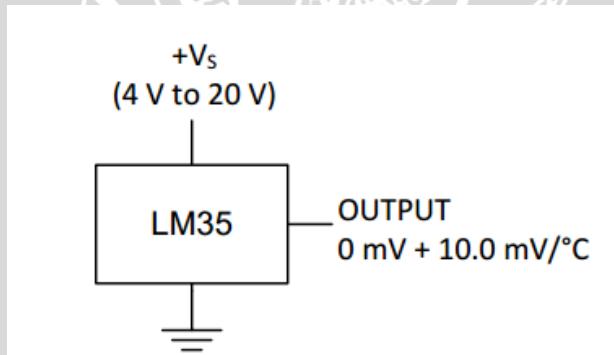
Sebuah *Software Serial Library* memungkinkan untuk berkomunikasi secara serial pada salah satu pin digital pada board Arduino mega 2560. Arduino mega 2560 juga mendukung I²C (TWI) dan komunikasi SPI. Perangkat lunak Arduino termasuk perpustakaan Kawat untuk menyederhanakan penggunaan bus I²C.

2.6 Sensor Suhu LM35

LM35 merupakan sensor suhu terintegrasi yang mempunyai tegangan keluaran yang linier. LM35 menghasilkan suatu keluaran tegangan yang proporsional dengan derajat Celcius. Tiga terminal aktifnya adalah tegangan catu daya (Vs), ground dan Vout. Tegangan keluaran LM35 adalah :

$$V_{out} = 10 \text{ mV/}^{\circ}\text{C} \quad (2.6)$$

Persamaan di atas menunjukkan bahwa untuk setiap kenaikan suhu 1°, tegangan keluaran meningkat sebesar 10 mV. Dalam Gambar 2.10 ditunjukkan skema sensor suhu LM35.

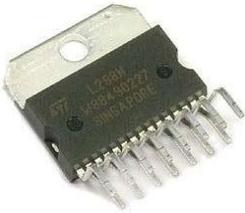


Gambar 2.10 Skema LM35 (Datasheet LM35)

2.7 Driver Motor DC

Motor DC tidak dapat dikendalikan secara langsung oleh mikrokontroler, karena kebutuhan arus listrik yang besar pada motor DC sedangkan arus keluaran pada mikrokontroler sangat kecil. *Driver* motor merupakan pilihan alternatif yang harus digunakan untuk mengendalikan kecepatan motor DC. Ada beberapa jenis kendali motor yang dapat digunakan untuk mengatur kecepatan motor, yaitu menggunakan rangkaian H-Bridge transistor, H-Bridge MOSFET, dan IC driver motor. Pada skripsi ini *driver* motor menggunakan IC driver L298N berbasis H-Bridge, yang mampu menangani beban hingga

4A pada tegangan 6V – 46V (Lihat Gambar 2.11). Dalam chip terdapat dua rangkaian H-Bridge. Selain itu driver ini mampu mengendalikan 2 motor sekaligus dengan arus beban masing-masing 2 A.



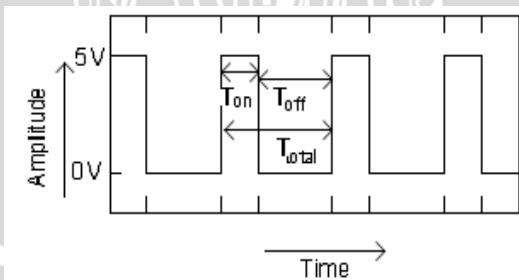
Gambar 2.11 IC *driver* motor L298N

Kecepatan motor DC dapat diatur dengan beberapa cara, yaitu dengan mengatur fluks medan, dengan mengatur tahanan jangkar, dan dengan mengatur tegangan sumber. Cara yang ketiga ini merupakan pengaturan yang sering digunakan karena penggunaannya yang relatif mudah. Pengaturan tegangan sumber biasanya menggunakan metode *Pulse Width Modulation* (PWM).

2.8 Sinyal PWM

Sinyal *Pulse Width Modulation* (PWM) adalah metode yang dapat digunakan untuk mengatur kecepatan dari motor DC. Dimana kecepatan motor DC tergantung pada besarnya *duty cycle* yang diberikan pada motor DC tersebut.

Pada sinyal PWM, frekuensi sinyal konstan sedangkan *duty cycle* bervariasi dari 0%-100%. *Duty cycle* adalah besarnya sinyal kontrol yang diberikan pada motor. Persamaan untuk perhitungan *duty cycle* ditunjukkan pada persamaan 2.8.1 dengan T_{on} adalah periode logika tinggi, dan T adalah periode keseluruhan. Sinyal PWM secara umum dapat dilihat dalam Gambar 2.12.



Gambar 2.12 Sinyal PWM secara umum

$$\text{Duty Cycle} = \frac{T_{on}}{T} \times 100\% \quad (2.8.1)$$

Sedangkan frekuensi sinyal dapat ditentukan dengan rumus sebagai berikut:

$$f_{OCn} = \frac{f_{clk_I/O}}{N \cdot 256} \quad (2.8.2)$$

Timer atau *counter* yang digunakan pada PWM ini yaitu *timer* atau *counter 0* (8 bit) dengan metode *fast PWM* dan *prescaler factor (N)* yaitu 256.

2.9 Liquid Crystal Display (LCD) Keypad Shield

LCD *Keypad Shield* dikembangkan untuk Arduino papan yang kompatibel, untuk menyediakan antarmuka yang *user-friendly* yang memungkinkan pengguna untuk pergi melalui menu, membuat pilihan dan lain-lain, terdiri dari 1602 karakter putih LCD dengan latar belakang biru. Tombol terdiri dari 5 tombol - *reset*, atas, kanan, bawah dan kiri. Untuk menyimpan pin IO digital, antarmuka *keypad* hanya menggunakan satu saluran ADC. Nilai tegangan dari setiap *keypad* pada saat membaca adalah 5 V. Contoh LCD *Keypad Shield* dapat dilihat dalam Gambar 2.13.



Gambar 2.13 LCD Keypad Shield (<http://www.jualarduino.com/wp-content/uploads/2013/09>)

BAB III METODE PENELITIAN

Untuk menyelesaikan rumusan masalah dan merealisasikan tujuan penelitian yang terdapat di bab pendahuluan maka diperlukan langkah-langkah untuk menyelesaikan masalah tersebut. Metode yang digunakan dapat diuraikan sebagai berikut.

3.1 Perancangan Sistem

Perancangan sistem dilakukan sebagai langkah awal sebelum terbentuknya suatu sistem, hal ini dimaksudkan agar sistem pengendalian kadar keasaman dapat berjalan sesuai deskripsi awal yang telah direncanakan. Perancangan sistem yang dilakukan meliputi :

1. Blok diagram sistem
2. Spesifikasi alat
3. Prinsip kerja sistem
4. Perancangan perangkat keras meliputi :
 - a. Perancangan kotak pengendali suhu
 - b. Perancangan rangkaian elektrik meliputi :
 - i. Rangkaian mikrokontroler
 - ii. Rangkaian *driver* motor
 - iii. Rangkaian *Liquid Crystal Display* (LCD)
5. Perancangan perangkat lunak meliputi:
 - a. Perancangan sistem secara keseluruhan
 - b. Perancangan sub rutin pembacaan sensor suhu
 - c. Perancangan sub rutin kontroler logika *fuzzy*
 - d. Perancangan sub rutin kecepatan kipas angin motor DC
 - e. Perancangan sub rutin tampilan *Liquid Crystal Display* (LCD)
 - f. Perancangan kontroler logika *fuzzy*

3.2 Pembuatan Sistem

Pembuatan sistem meliputi pembuatan perangkat keras (*hardware*) dan pembuatan perangkat lunak (*software*) sebagai berikut :

1. Pembuatan perangkat keras (*hardware*)

Pembuatan perangkat keras (*hardware*) meliputi perancangan rangkaian mikrokontroler, rangkaian *driver* motor dan rangkaian LCD.

2. Pembuatan perangkat lunak (*software*)



Pembuatan perangkat lunak (*software*) yaitu pembuatan program dengan *software* program Arduino 1.0.6 berdasarkan *flowchart* yang telah dirancang.

3.3 Pengujian dan Analisa Data

Pengujian sistem pengendalian suhu bertujuan untuk menguji apakah sistem ini bekerja sesuai dengan perancangan. Pengujian sistem ini meliputi pengujian setiap blok rangkaian dan pengujian keseluruhan sistem.

1. Pengujian penentuan *setpoint*
2. Pengujian setiap blok rangkaian, meliputi :
 - a. Pengujian *Liquid Crystal Display* (LCD)
 - b. Pengujian sensor suhu
 - c. Pengujian *driver* motor
3. Pengujian laptop tanpa sistem pengendali
4. Pengujian keseluruhan sistem

Pengujian keseluruhan sistem dilakukan dengan menyambungkan seluruh blok perangkat keras maupun perangkat lunak yang telah dibuat. Sistem dikatakan berhasil jika sistem telah bekerja sesuai dengan spesifikasi rancangan.

3.4 Pengambilan Kesimpulan dan Saran

Kesimpulan diperoleh berdasarkan data pengujian sistem secara keseluruhan. Jika hasil yang diperoleh sesuai dengan yang direncanakan, maka sistem kendali tersebut telah memenuhi harapan dan dapat dikembangkan untuk penelitian selanjutnya.



BAB IV PERANCANGAN DAN PEMBUATAN ALAT

Pada bab ini akan dibahas perancangan dan pembuatan alat. Pembahasan akan dilakukan pada setiap blok rangkaian, cara kerja masing-masing blok rangkaian, perhitungan dan juga fungsi masing-masing blok rangkaian.

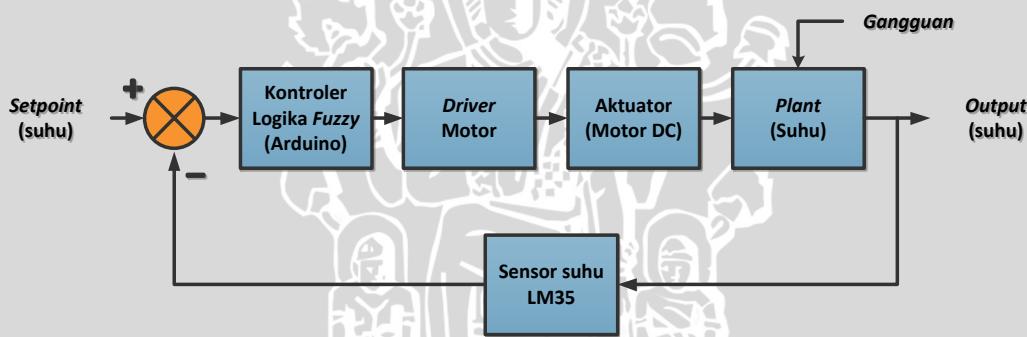
Secara garis besar terdapat dua bagian perangkat yang ada yaitu :

1. Perancangan perangkat keras (*hardware*)
2. Perancangan perangkat lunak (*software*)

Pada perancangan perangkat keras meliputi perangkat elektrik maupun mekanik yang digunakan pada sistem pengendalian suhu. Sedangkan pada perancangan perangkat lunak meliputi *flowchart* dan *software*.

4.1 Diagram Blok Sistem

Diagram blok sistem yang dirancang ditunjukkan dalam Gambar 4.1.



Gambar 4.1 Blok diagram sistem

Keterangan :

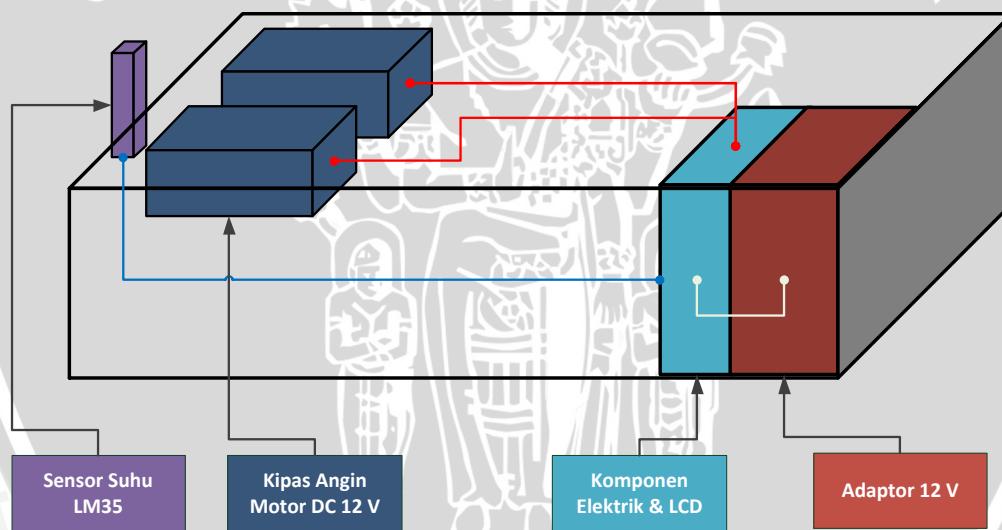
1. *Setpoint* sistem berupa suhu 49°C (diperoleh setelah dilakukan pengujian pada bab 5).
2. Kontroler yang digunakan adalah kontroler logika *fuzzy* yang diprogram ke dalam Arduino Mega.
3. Aktuator yang digunakan adalah kipas angin motor DC yang berfungsi untuk membuang suhu panas dari dalam laptop.
4. Sebagai umpan balik digunakan Sensor Suhu untuk mengukur suhu yang dibuang oleh Laptop.

4.2 Spesifikasi Alat

Spesifikasi komponen-komponen sistem pengendalian kadar keasaman menggunakan Arduino Mega adalah sebagai berikut :

1. Kotak yang digunakan terbuat dari bahan plastik dengan ukuran 35 cm x 27 cm x 7 cm.
2. Adaptor yang digunakan memiliki sumber tegangan sebesar 12 V dan arus 1200 mA.
3. Perangkat kontrol yang digunakan adalah Arduino Mega.
4. Driver Motor yang digunakan yaitu IC L298N.
5. Sensor suhu yang digunakan yaitu sensor suhu LM35.
6. Aktuator yang digunakan adalah kipas angin motor DC dengan tegangan maksimal 12 V.

Skema perancangan sistem pengendalian suhu pada prosesor menggunakan KLF ditunjukkan dalam Gambar 4.2.



Gambar 4.2 Skema perancangan sistem pengendalian suhu pada prosesor

4.3 Prinsip Kerja Sistem

Cara kerja sistem pengendalian suhu pada prosesor menggunakan Kontrol Logika Fuzzy sebagai berikut :

1. Sistem diberi catu daya. Catu daya sebesar 5 V digunakan untuk mencatu rangkaian LCD, Mikrokontroler Arduino Mega dan *driver* motor . Dan catu daya 12 V digunakan untuk mencatu kipas angin motor DC.
2. Pengukuran suhu pada laptop menggunakan sensor suhu.
3. Sensor suhu dihubungkan Arduino Mega dengan rentang 0-5 V.

4. Keluaran Arduino Mega berupa sinyal *pulse width modulation* (PWM) diberikan ke *driver* motor. *Driver* berfungsi untuk menguatkan sinyal *Pulse Width Modulation* (PWM) Arduino Uno dari tegangan sebesar 0-5 V menjadi 0-12 V.
5. Sistem dari alat ini bekerja sesuai dengan penilaian dari sensor suhu yang mengukur besaran suhu yang keluar dari udara yang terbuang dari dalam laptop. Setelah suhu didapatkan, maka kontroler akan memberikan perintah kepada kipas angin motor DC untuk berputar.
6. Sensor suhu LM35 akan mengukur secara berkala, jika suhu output melebihi *setpoint* maka Arduino akan memerintahkan pada Motor DC untuk berputar lebih cepat. Dengan demikian suhu prosesor pada laptop diharapkan akan berada pada Suhu *setpoint*.

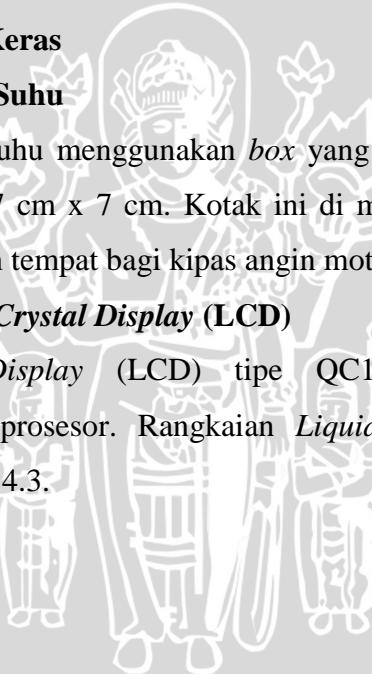
4.4 Perancangan Perangkat Keras

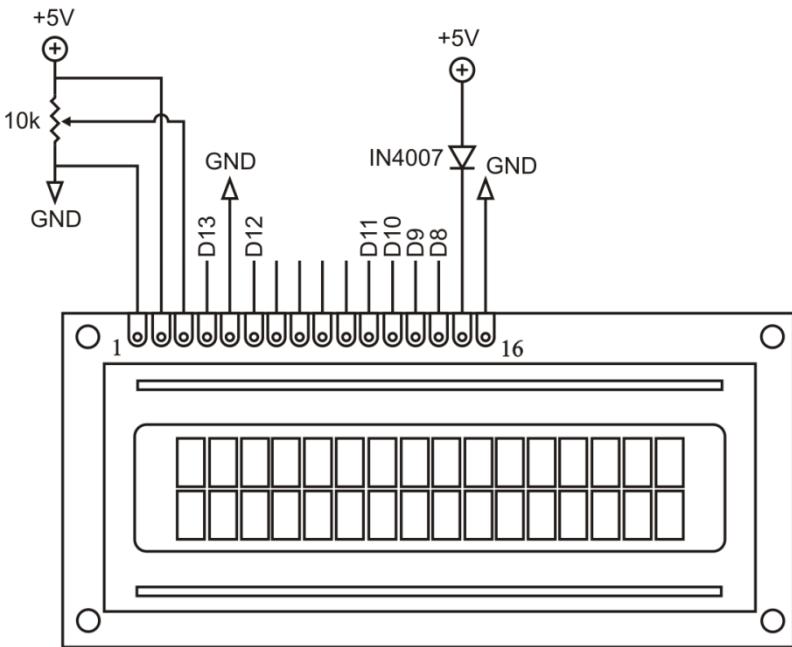
4.4.1 Kotak Pengendali Suhu

Kotak pengendali suhu menggunakan *box* yang terbuat dari bahan plastik dengan ukuran 35 cm x 27 cm x 7 cm. Kotak ini di modifikasi sedemikian rupa sehingga dapat memberikan tempat bagi kipas angin motor DC.

4.4.2 Rangkaian *Liquid Crystal Display* (LCD)

Liquid Crystal Display (LCD) tipe QC1602A digunakan untuk menampilkan suhu pada prosesor. Rangkaian *Liquid Crystal Display* (LCD) ditunjukkan dalam Gambar 4.3.





Gambar 4.3 Rangkaian Liquid Crystal Display (LCD)

Keterangan :

1. Pin 1 atau pin VSS dihubungkan dengan *ground*.
2. Pin 2 atau pin VDD dihubungkan dengan +5V.
3. Pin 5 atau pin R/W dihubungkan dengan *ground*.
4. Pin 4 atau pin RS dihubungkan dengan pin *Digital 32* Arduino Uno.
5. Pin 6 atau pin E dihubungkan dengan pin *Digital 30* Arduino Uno.
6. Pin 11 atau pin DB4 dihubungkan dengan pin *Digital 28* Arduino Uno.
7. Pin 12 atau pin DB5 dihubungkan dengan pin *Digital 26* Arduino Uno.
8. Pin 13 atau pin DB6 dihubungkan dengan pin *Digital 24* Arduino Uno.
9. Pin 14 atau pin DB7 dihubungkan dengan pin *Digital 22* Arduino Uno.
10. Pin 16 atau pin LEDK dihubungkan dengan *ground*.
11. Pin 15 atau pin LEDA dihubungkan dengan diode 1N4007.

4.4.3 Rangkaian Driver Motor

Driver motor yang digunakan adalah IC L298N. *Driver* motor ini digunakan untuk mengendalikan motor DC yaitu kipas angin motor DC. Mode pengendalian kecepatan motor yang digunakan adalah mode sinyal *Pulse Width Modulation* (PWM). Catu daya motor yang digunakan adalah catu daya eksternal 12 V.



Konfigurasi pin *driver motor* ini ditunjukkan dalam Tabel 4.1.

Tabel 4.1 Konfigurasi pin *driver motor* pada Arduino Mega

Pin	Pin pada Arduino	Fungsi
5	9 (PWM)	Kontrol arah motor 1
7	10 (PWM)	Kontrol arah motor 1
6	8 (PWM)	Kontrol PWM motor 1
9	5 V	<i>Supply</i> tegangan 5 V

4.4.4 Rangkaian Mikrokontroler

Sistem pengendalian suhu ini menggunakan Arduino Mega sebagai pusat pengolah utama dalam proses pengendalian. Konfigurasi pin dari Arduino Mega ditunjukkan dalam Tabel 4.2.

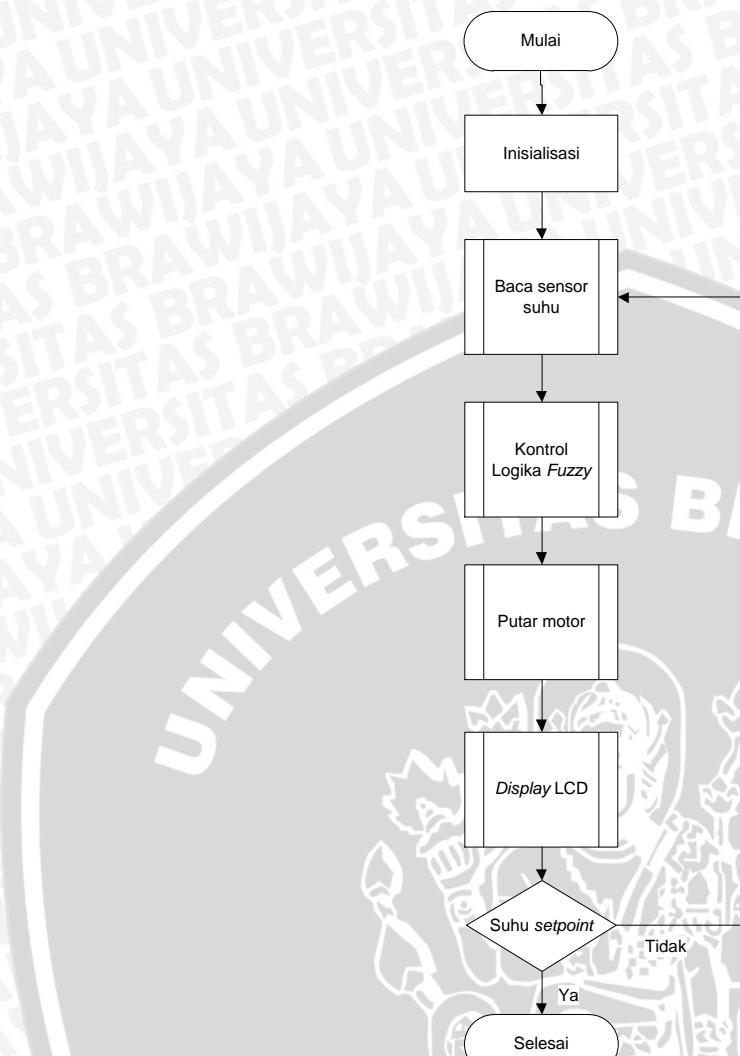
Tabel 4.2 Konfigurasi pin Arduino Uno

No	Pin	Fungsi
1	A0	Sensor suhu LM35
2	PWM 8	<i>Enable</i> (<i>driver motor</i>)
3	PWM 9	<i>Input 1</i> (<i>driver motor</i>)
4	PWM 10	<i>Input 2</i> (<i>driver motor</i>)
5	Vin	<i>Input 5 V</i> (<i>adaptor</i>)
6	GND	VSS, R/W, LEDK (<i>adaptor</i> & LCD)
7	D22	DB7 (LCD)
8	D24	DB6 (LCD)
9	D26	DB5 (LCD)
10	D28	DB4 (LCD)
11	D30	E (LCD)
12	D32	RS (LCD)

4.5 Perancangan Perangkat Lunak

4.5.1 *Flowchart* Sistem Secara Keseluruhan

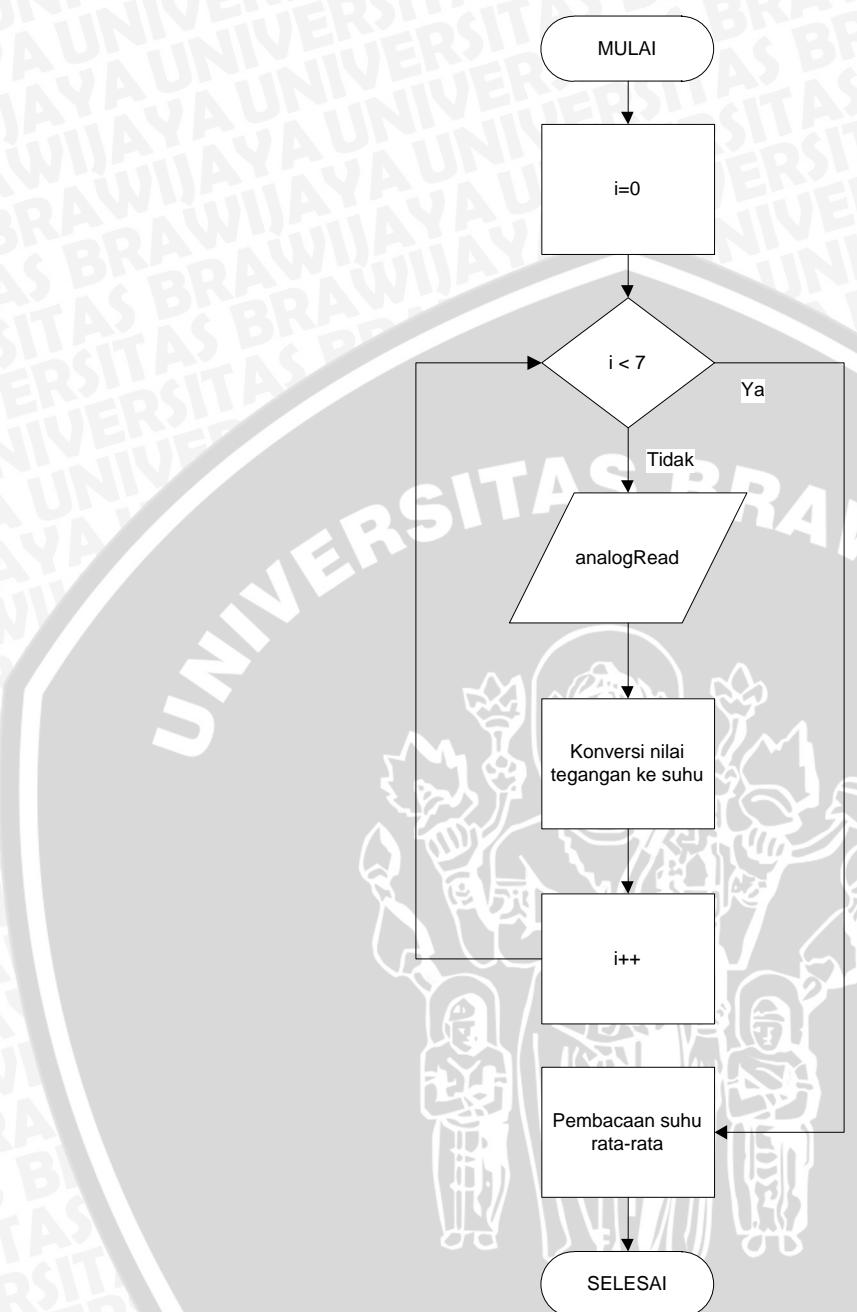
Flowchart keseluruhan sistem ditunjukkan dalam Gambar 4.4.



Gambar 4.4 *Flowchart* keseluruhan sistem

4.5.2 *Flowchart* Pembacaan Sensor suhu

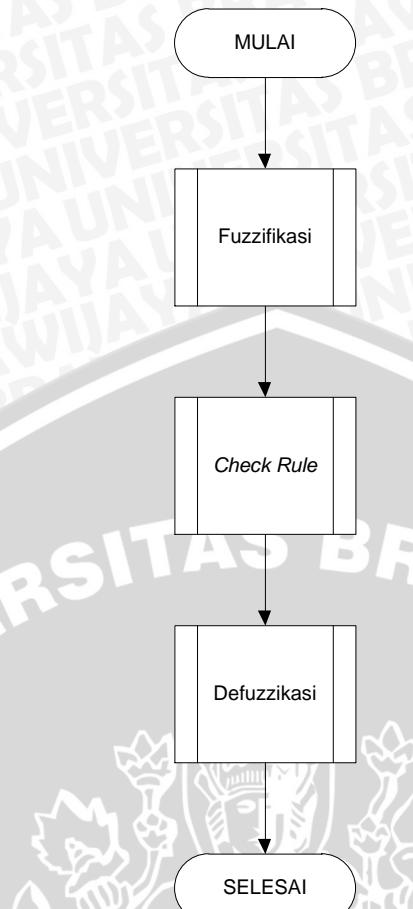
Flowchart pembacaan sensor suhu ditunjukkan dalam Gambar 4.5.



Gambar 4.5 Flowchart pembacaan sensor suhu

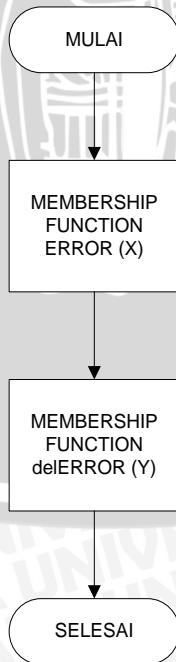
4.5.3 Flowchart Kontroler Logika Fuzzy

Flowchart kontroler logika fuzzy ditunjukkan dalam Gambar 4.6.



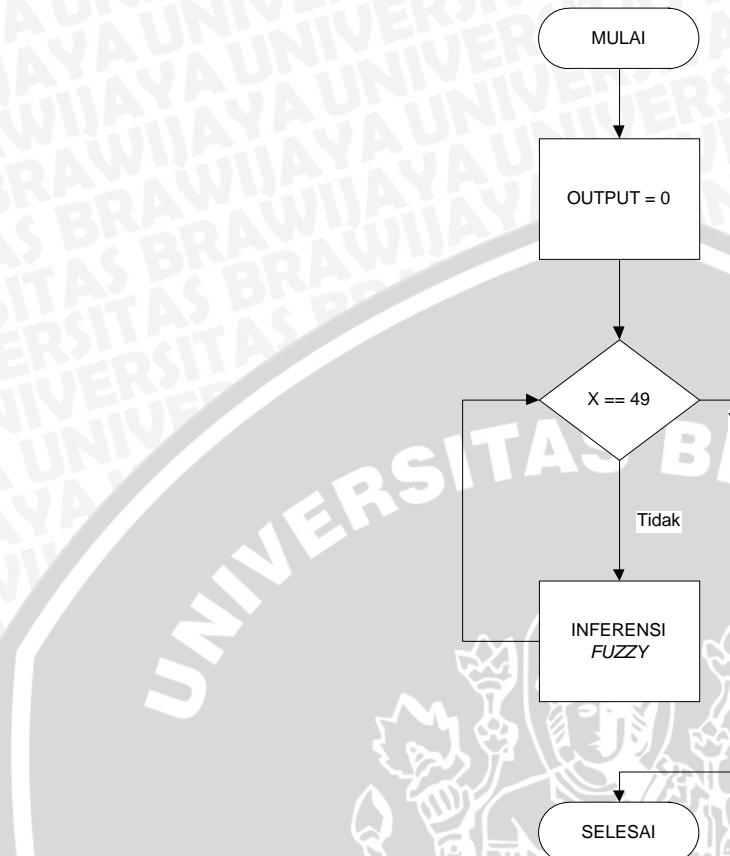
Gambar 4.6 Flowchart kontroler logika fuzzy

Flowchart sub rutin fuzzy ditunjukkan dalam Gambar 4.7.



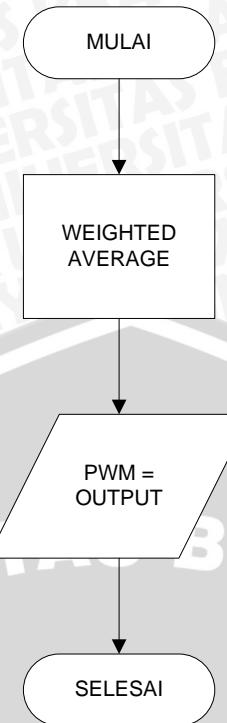
Gambar 4.7 Flowchart subrutin fuzzy

Flowchart sub rutin *check rule* ditunjukkan dalam Gambar 4.8.



Gambar 4.8 Flowchart subrutin *check rule*

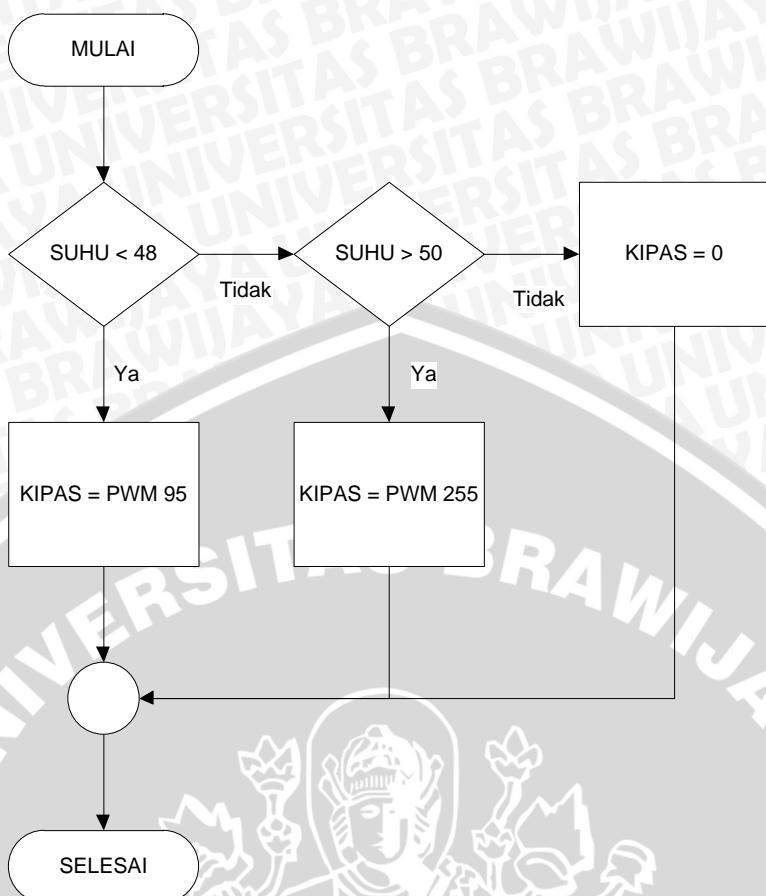
Flowchart sub rutin *defuzzy* ditunjukkan dalam Gambar 4.9.



Gambar 4.9 *Flowchart* subrutin defuzzy

4.5.4 *Flowchart Putaran Motor*

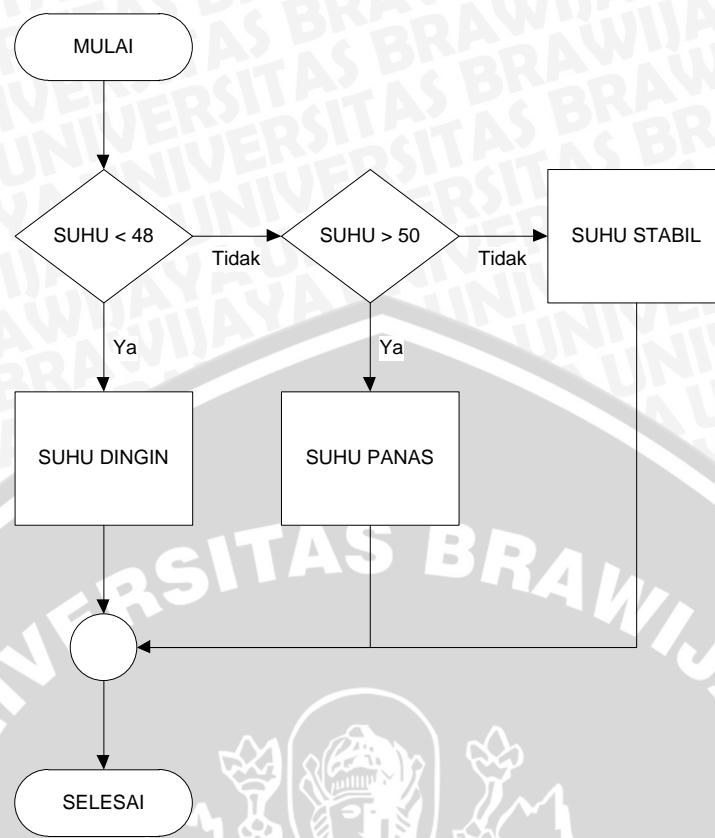
Flowchart putaran motor ditunjukkan dalam Gambar 4.10.



Gambar 4.10 Flowchart putaran motor

4.5.5 Flowchart Tampilan Liquid Crystal Display (LCD)

Flowchart tampilan *Liquid Crystal Display* (LCD) ditunjukkan dalam Gambar 4.11.



Gambar 4.11 Flowchart Tampilan Liquid Crystal Display (LCD)

4.5.6 Perancangan Kontroler Logika Fuzzy

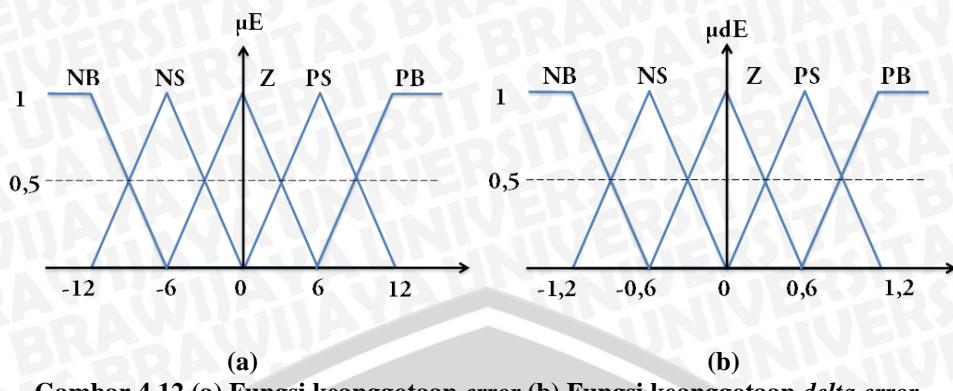
Pada perancangan kontroler logika *fuzzy* dilakukan langkah-langkah sebagai berikut :

4.5.6.1 Menentukan fungsi keanggotaan masukan dan keluaran

- Fungsi keanggotaan masukan.

Variabel yang digunakan untuk masukan adalah *error* dan *delta error*. *Error* adalah nilai *setpoint* dikurangi nilai sebenarnya, sedangkan *delta error* adalah nilai *error* sekarang dikurangi *error* sebelumnya. Fungsi keanggotaan dari *error* dan *delta error* terdiri atas 5 label, yaitu *Negative Big* (NB), *Negative Small* (NS), *Zero* (Z), *Positive Small* (PS), dan *Positive Big* (PB). Data masukan dari fungsi keanggotaan diperoleh dari nilai pengukuran suhu oleh sensor suhu.

Fungsi keanggotaan berjumlah 5 dengan tujuan agar menghasilkan aturan *fuzzy* yang banyak sehingga semakin banyak pula kondisi dan aksi kontrol yang akan dipilih. Fungsi keanggotaan masukan ditunjukkan dalam Gambar 4.12.

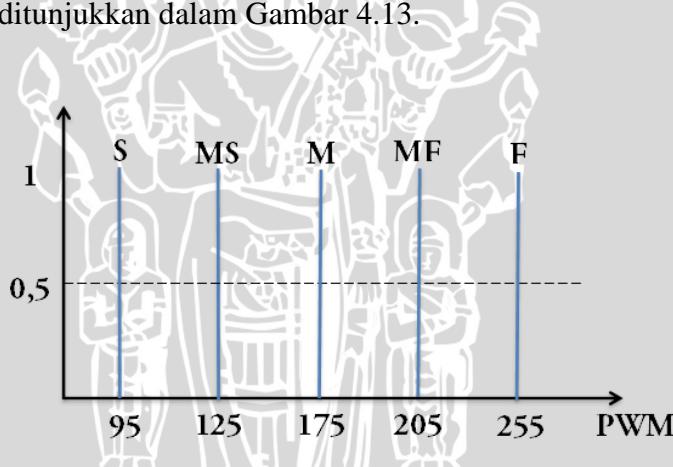


Gambar 4.12 (a) Fungsi keanggotaan *error* (b) Fungsi keanggotaan *delta error*

b. Fungsi keanggotaan keluaran.

Fungsi keanggotaan keluaran motor pompa DC merupakan nilai PWM yang dikeluarkan oleh Arduino Mega berupa hasil perhitungan dengan metode defuzzifikasi *Center of Gravity* (COG) yang telah dibulatkan dalam program.

Fungsi keanggotaan keluaran berjumlah 5 yaitu *Slow* (S), *Medium Slow* (MS), *Medium* (M), *Medium Fast* (MF) dan *Fast* (F) yang ditunjukkan dalam Gambar 4.13.



Gambar 4.13 Fungsi keanggotaan keluaran

4.5.7.2 Menentukan aturan fuzzy

Aturan *fuzzy* (*fuzzy rule*) digunakan sebagai penentu keluaran dari fuzzifikasi yang akan diolah dalam proses defuzzifikasi. Jumlah aturan *fuzzy* yang digunakan sebanyak 25. Aturan *fuzzy* diperoleh dari 5 fungsi keanggotaan masukan *error* dan *delta error*. Aturan *fuzzy* yang digunakan ditunjukkan dalam Tabel 4.3.



Tabel 4.3 Aturan Fuzzy pada Sistem

		Error				
		NB	NS	Z	PS	PB
Delta Error	NB	S	S	S	M	M
	NS	S	S	S	M	MF
	Z	S	S	MS	MF	MF
	PS	S	MS	MS	MF	F
	PB	MS	MS	M	F	F

4.5.7.3 Menentukan metode inferensi fuzzy.

Metode inferensi fuzzy yang digunakan adalah metode inferensi MAX-MIN.

4.5.7.4 Menentukan metode defuzzifikasi.

Defuzzifikasi merupakan proses untuk mengubah keluaran fuzzy menjadi keluaran *crisp*. Hasil defuzzifikasi digunakan untuk mengatur kecepatan kipas angin motor DC. Metode defuzzifikasi yang digunakan adalah COG.

BAB V

PENGUJIAN DAN ANALISIS

Tujuan pengujian sistem adalah untuk menentukan apakah alat yang telah dibuat berfungsi dengan baik dan sesuai dengan perancangan. Pengujian pada sistem ini meliputi pengujian setiap blok maupun pengujian secara keseluruhan. Pengujian setiap blok ini dilakukan untuk menemukan letak kesalahan dan mempermudah analisis pada sistem apabila alat tidak bekerja sesuai dengan perancangan.

5.1 Pengujian Penentuan Setpoint

a. Tujuan

Mengetahui suhu prosesor laptop yang membutuhkan kecepatan putaran kipas internal paling lambat. Pembacaan sensor pada prosesor laptop dihitung dengan mengukur rata-rata pembacaan pada setiap *core* pada prosesor. Peralatan yang digunakan :

- Laptop
- Program SpeedFan 4.47

b. Langkah Pengujian

1. Menyalakan laptop dan membuka Program SpeedFan.
2. Mengamati dan mencatat setiap perubahan suhu dan kecepatan kipas internal.

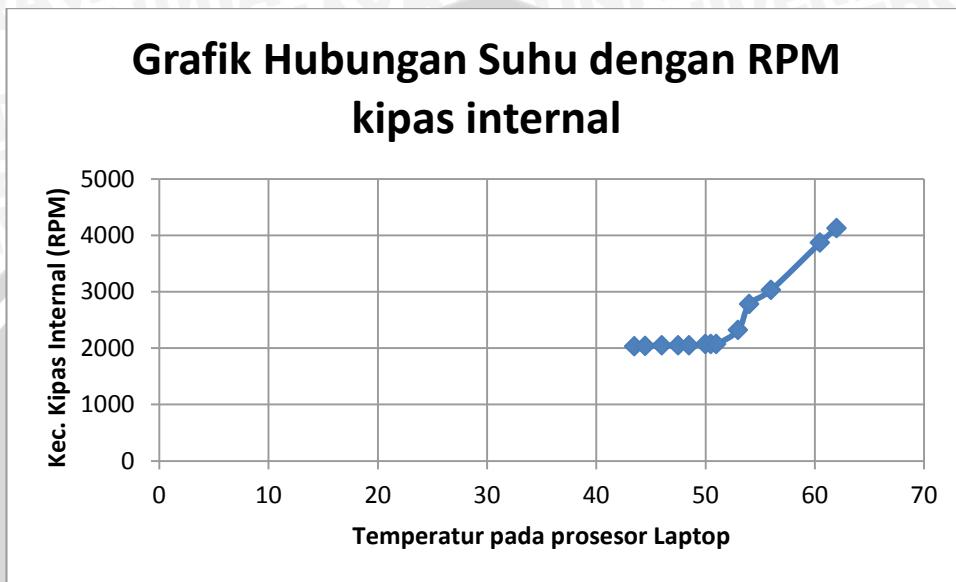
c. Hasil Pengujian dan Analisis

Hasil pengujian suhu prosesor laptop ditunjukkan dalam Tabel 5.1. Didapatkan grafik linearitas antara suhu dengan kecepatan kipas internal seperti yang ditunjukkan dalam Gambar 5.1.

Tabel 5.1 Hasil Pengujian Suhu Prosesor Laptop

Core 1 (°C)	Core 2 (°C)	Rata-rata (°C)	RPM
44	43	43,5	2030
45	44	44,5	2040
46	46	46	2050
47	48	47,5	2050
49	48	48,5	2050
51	49	50	2070
51	50	50,5	2070
52	50	51	2070

53	53	53	2320
54	54	54	2780
56	56	56	3030
61	60	60,5	3870
62	62	62	4130



Gambar 5.1 Grafik Hubungan Suhu dengan RPM kipas internal

Sesuai dengan hasil pengujian pada Tabel 5.1 dan Gambar 5.2, ditentukan *setpoint* dengan suhu sebesar 49 °C, karena suhu tersebut adalah suhu paling tinggi untuk kipas internal berputar dengan kecepatan +-2000 RPM. Dan pada suhu itu pula kecepatan kipas internal dapat berputar kembali dengan kecepatan +-2000 RPM jika kipas internal telah melebihi kecepatan 3000 RPM.

5.2 Pengujian LCD

a. Tujuan

Mengetahui bahwa masing-masing pin mikrokontroler dapat mengirim data menuju LCD sesuai dengan program yang dibuat dan LCD dapat menampilkan data karakter dengan benar.

b. Peralatan yang Digunakan

- Mikrokontroler Arduino Mega
- Catu daya 5 V
- LCD



- c. Langkah Pengujian
 - 1. Menghubungkan LCD dengan Arduino Mega (sesuai perancangan perangkat keras pada bab 4).
 - 2. Memberikan catu daya pada Arduino Mega
 - 3. Membuat dan mengunduh program tampilan nama pada software Arduino 1.0.6.
- d. Hasil Pengujian dan Analisis

Hasil pengujian dapat dilihat dalam Gambar 5.2.



Gambar 5.2 Tampilan Data Karakter pada LCD

5.3 Pengujian Sensor LM35

a. Tujuan

Mengetahui kemampuan pembacaan sensor LM35 terhadap pembacaan sensor suhu di dalam laptop dengan cara mengamati dan mencatat setiap perubahan suhu yang terjadi pada suhu prosesor dan membandingkannya dengan nilai suhu yang didapatkan oleh sensor suhu LM35.

b. Peralatan yang Digunakan

- Sensor LM35
- Mikrokontroler Arduino Mega
- Catu daya 5 V
- Laptop Asus A42JC
- Program SpeedFan 4.47

c. Langkah Pengujian

- 1. Menghubungkan sensor suhu LM35 dengan Arduino Mega (sesuai perancangan perangkat keras pada bab 4).
- 2. Menyalakan laptop dan memberikan catu daya pada Arduino Mega.

Lalu membuka Program SpeedFan pada laptop.

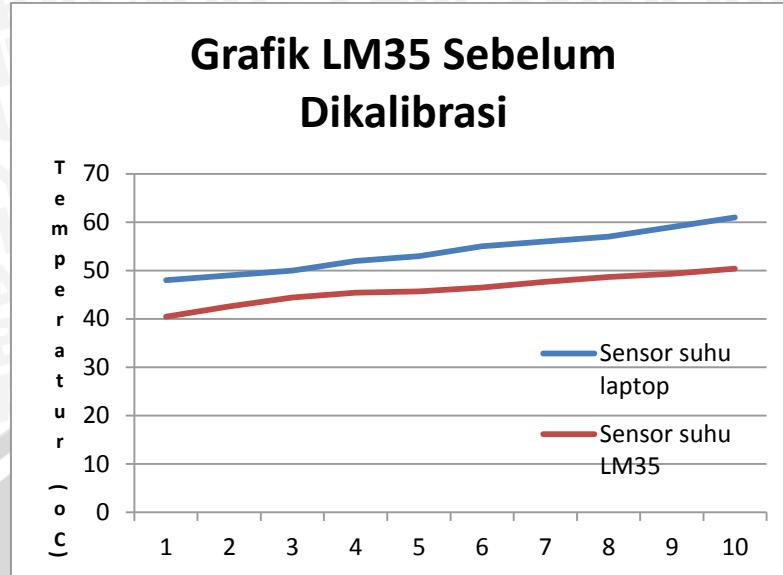
3. Meletakkan sensor suhu LM35 di dekat ventilasi pembuangan udara laptop.
 4. Membuat dan mengunduh program termometer digital pada *software* Arduino 1.0.6.
 5. Mengamati dan mencatat setiap perubahan suhu berdasarkan suhu prosesor.
- d. Hasil Pengujian dan Analisis

Hasil pengujian sensor suhu LM35 ditunjukkan dalam Tabel 5.2.

Tabel 5.2 Hasil Pengujian Sensor LM35 Sebelum Dikalibrasi

Sensor suhu di dalam leptop (°C)	Sensor suhu LM35 (°C)	Error (%)
48	40,49	15,65%
49	42,56	13,14%
50	44,46	11,08%
52	45,4	12,69%
53	45,72	13,74%
55	46,51	15,44%
56	47,69	14,84%
57	48,64	14,67%
59	49,35	16,36%
61	50,38	17,41%
Rata-rata error		14,73%

Berdasarkan hasil pengujian pada Tabel 5.2, rata-rata *error* antara pembacaan suhu prosesor dengan pembacaan suhu LM35 masih sangat besar yaitu 14,73%. Pada grafik 5.3 terlihat bahwa selisih yang timbul antara kedua pembacaan suhu cenderung membentuk garis linear. Sehingga untuk mengkompensasi *error* pembacaan suhu, dapat dikalibrasi menggunakan metode regresi linear.



Gambar 5.3 Grafik Perbandingan Suhu Prosesor dengan Suhu LM35

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

Tabel 5.3 Hasil Pengujian Sensor LM35 Setelah Dikalibrasi

Sensor suhu di dalam leptop (°C)	Sensor suhu LM35 (°C)	Error (%)
48	45,85	4,47%
49	48,85	0,29%
50	51,61	-3,23%
52	52,97	-1,88%
53	53,44	-0,84%
55	54,59	0,74%
56	56,30	-0,54%
57	57,68	-1,20%
59	58,71	0,49%
61	60,20	1,30%
Rata-rata <i>error</i>		1,5 %

Berdasarkan Tabel 5.3, hasil pengujian sensor LM35 menghasilkan pembacaan suhu yang mendekati suhu prosesor dengan *error* rata-rata sebesar 2,26%, sehingga kalibrasi yang dilakukan telah sesuai dan sensor LM35 dapat bekerja dengan baik.

5.4 Pengujian Driver Motor DC

- a. Tujuan

Mengetahui pengaruh perubahan pulsa PWM terhadap tegangan keluaran pada pin Arduino Mega dan pin IC L298N.

b. Peralatan yang Digunakan

- Mikrokontroler Arduino Mega
- Catu daya 5 V
- Rangkaian Driver Motor L298N
- Multimeter
- LCD
- *Potentiometer*

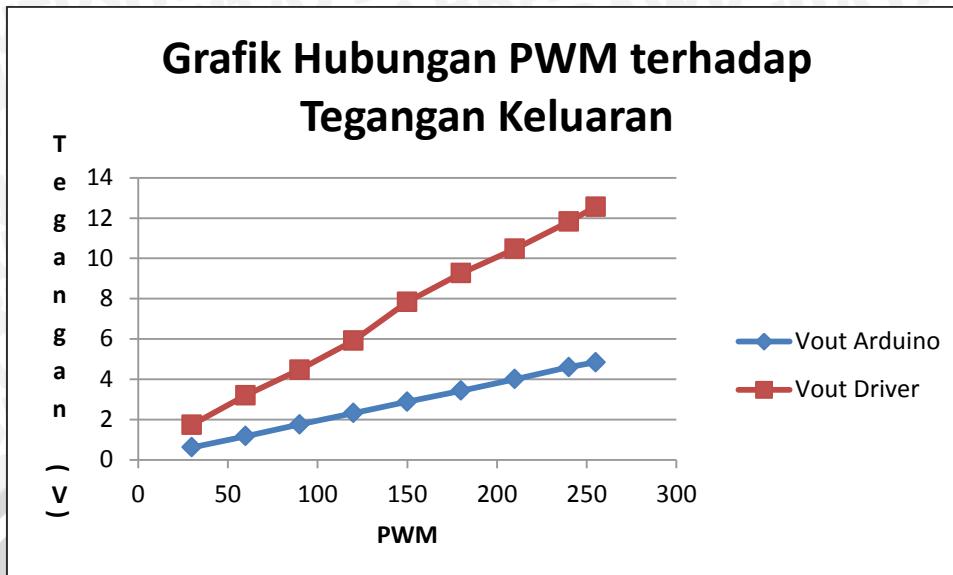
c. Langkah Pengujian

1. Menghubungkan Arduino Mega dengan IC L298N dan LCD (sesuai perancangan perangkat keras pada bab 4).
2. Menghubungkan *Potentiometer* dengan Arduino Mega pada pin analog, sebagai pengendali sinyal PWM yang dikeluarkan oleh pin Arduino Mega.
3. Memberikan catu daya pada Arduino Mega.
4. Membuat dan mengunduh program pengatur PWM pada *software* Arduino 1.0.6
5. Mengamati dan mencatat perubahan tegangan keluaran pada pin keluaran Arduino Mega dan pin IC L298N terhadap perubahan PWM.

d. Hasil Pengujian dan Analisis

Hasil pengujian sinyal PWM terhadap tegangan keluaran dapat dilihat dalam Gambar 5.4 yang merupakan hubungan PWM terhadap tegangan keluaran. Data untuk Gambar 5.3 dapat dilihat dalam Tabel 5.4.





Gambar 5.4 Grafik Hubungan PWM terhadap Sudut Motor Servo

Tabel 5.4 Hasil Pengujian Driver Motor L298N

PWM	Tegangan pada Arduino (V)	Tegangan pada Driver (V)
0	0	0
30	0,62	1,74
60	1,17	3,2
90	1,75	4,47
120	2,32	5,91
150	2,88	7,84
180	3,43	9,26
210	4	10,47
240	4,6	11,82
255	4,84	12,56

Hasil pengujian menunjukkan bahwa driver motor dapat mengeluarkan tegangan diatas 5 V dan besar tegangan dapat dikontrol oleh sinyal PWM yang diberikan oleh Arduino Mega.

5.5 Pengujian Laptop Tanpa Sistem Pengendali

a. Tujuan

Pengujian ini bertujuan untuk mengetahui bagaimana respon laptop terhadap *setpoint* ketika laptop bekerja tanpa sistem pengendali, sehingga dapat ditentukan bahwa laptop memerlukan sebuah sistem pengendalian.

b. Peralatan yang digunakan

- Laptop

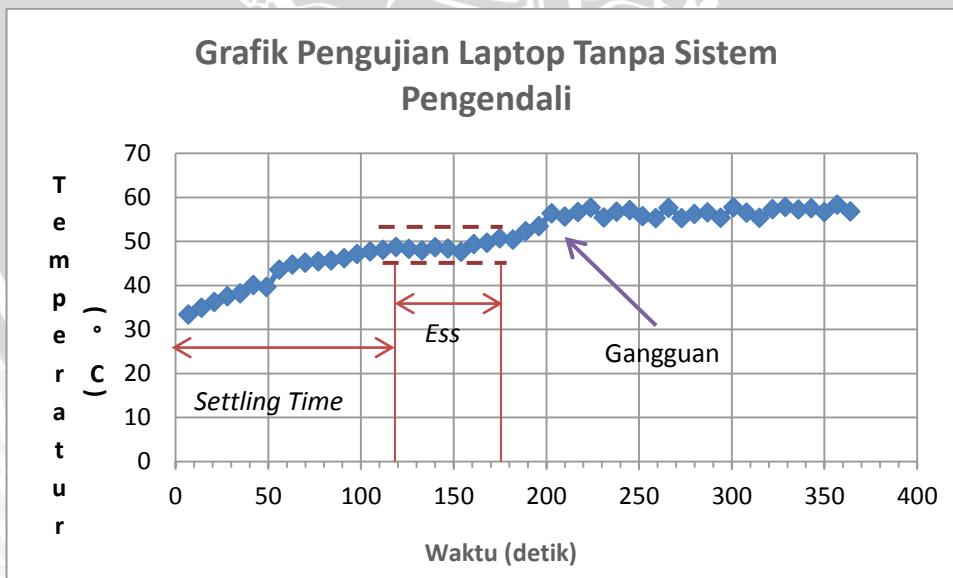
- Arduino Mega
- Sensor LM35
- Catu daya 5 V
- LCD
- Laptop
- Program dan *software* Arduino 1.0.6.

c. Langkah Pengujian

1. Menghubungkan Sensor LM35 dan LCD ke Arduino Mega.
2. Memberikan catu daya pada Arduino Mega.
3. Mengunduh program pembaca temperatur pada *software* Arduino 1.0.6.
4. Menampilkan temperatur pada LCD.
5. Mengamati dan mencatat temperatur yang tampil pada LCD, lalu membuat grafik temperatur terhadap waktu.

d. Hasil Pengujian

Setelah melakukan prosedur pengujian, didapatkan hasil respon temperatur prosesor laptop. Grafik hasil pengujian ditunjukkan pada Gambar 5.5.



Gambar 5.5 Pengujian Laptop tanpa Sistem Pengendali

Dari grafik hasil pengujian, dapat diketahui bahwa prosesor laptop mampu mencapai *setpoint* pada suhu 49 °C setelah 119 detik. Dan prosesor laptop memiliki nilai *error steady state* sebesar :

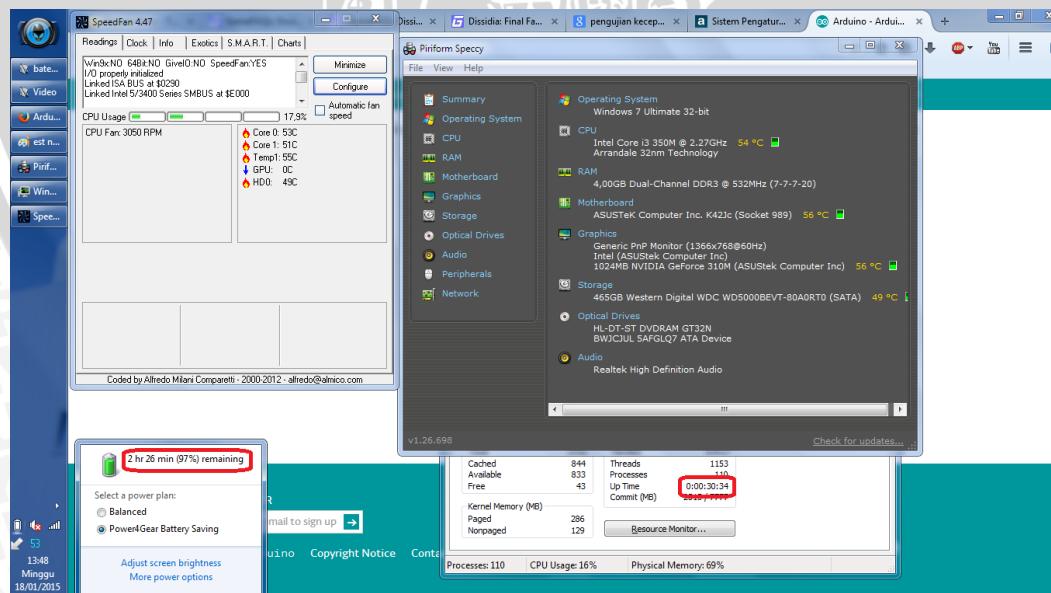
$$\% \text{ Error steady state} =$$

$$\begin{aligned} &= \frac{1}{N} \cdot \sum_{i=1}^N \frac{|PV(i) - 49|}{49} \cdot 100\% \\ &= 0,847 / 49 \cdot 100\% \\ &= 1,72\% \end{aligned}$$

Dan dengan mendapat gangguan berupa penambahan program *browsing* (Mozilla / Chrome), prosesor laptop mendapat *error* terbesar sebesar :

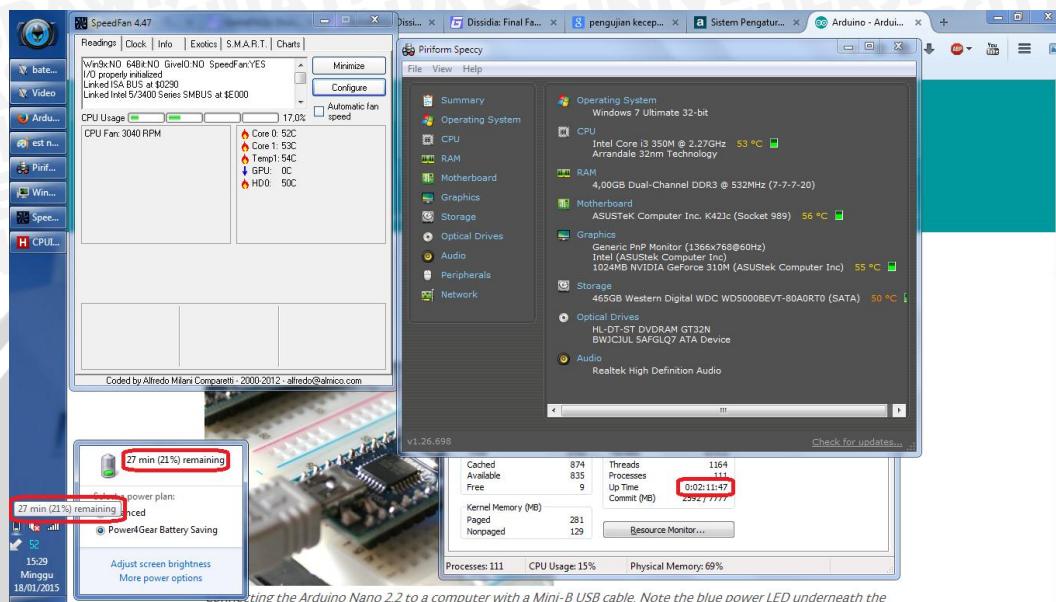
$$\begin{aligned} \% \text{ Max error dengan gangguan} &= \frac{|58,32 - 49|}{49} \times 100\% \\ &= 0,1902 \times 100\% \\ &= 19,02\% \end{aligned}$$

Dalam pengujian ini dapat disimpulkan bahwa laptop tidak dapat kembali setelah diberikan gangguan, maka dari itu laptop perlu diberikan sistem pengendali agar respon yang dihasilkan dapat mendekati *setpoint* yang diberikan. Untuk pengujian waktu pemakaian baterai laptop dapat dilihat dalam Gambar 5.6.



Gambar 5.6 Pemakaian Awal Baterai Laptop tanpa Sistem Pengendali

Dalam Gambar 5.6 tertera estimasi waktu pemakaian laptop dengan menggunakan baterai selama 2 jam 26 menit. Kecepatan kipas internal yang digunakan setelah diberi gangguan rata-rata sebesar 3000 RPM. Untuk pemakaian akhir saat baterai mencapai 21 % dapat dilihat dalam Gambar 5.7.



Gambar 5.7 Pemakaian Akhir Baterai Laptop tanpa Sistem Pengendali

Dalam Gambar 5.6 dan 5.7 dapat disimpulkan bahwa waktu pemakaian aktual baterai mulai dari kapasitas 97 % s/d 21 %, adalah 101 menit atau 1 jam 40 menit.

5.6 Pengujian Sistem Secara Keseluruhan

a. Tujuan

Pengujian ini bertujuan untuk mengetahui bagaimana kinerja sistem secara keseluruhan dan mengamati respon KLF sebelum mendapatkan gangguan berupa penambahan program.

b. Peralatan yang digunakan

- Laptop
- Arduino Mega
- Sensor LM35
- Catu daya 5 V
- LCD
- Kipas angin motor DC

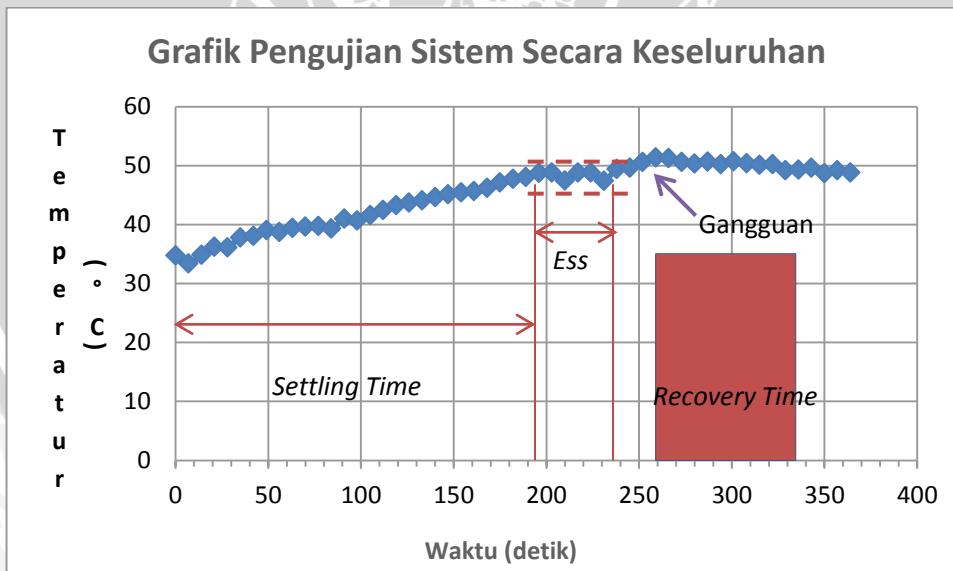
- IC L298N
- Laptop
- Program dan *software* Arduino 1.0.6.

c. Langkah Pengujian

1. Menghubungkan Sensor LM35, LCD, kipas angin motor DC dan IC L298N ke Arduino Mega.
2. Memberikan catu daya pada Arduino Mega.
3. Membuat dan mengunduh program pengendali suhu pada *software* Arduino 1.0.6.
4. Menampilkan temperatur pada LCD.
5. Mengamati dan mencatat temperatur yang tampil pada LCD, lalu membuat grafik temperatur terhadap waktu.

d. Hasil Pengujian

Setelah melakukan prosedur pengujian, didapatkan hasil respon temperatur prosesor laptop. Grafik hasil pengujian ditunjukkan dalam Gambar 5.8.



Gambar 5.8 Pengujian Sistem Secara Keseluruhan

Dari grafik hasil pengujian, sistem dapat mencapai *setpoint* dengan *settling time* sebesar 196 detik. Dan sistem mendapat *error steady state* sebesar :

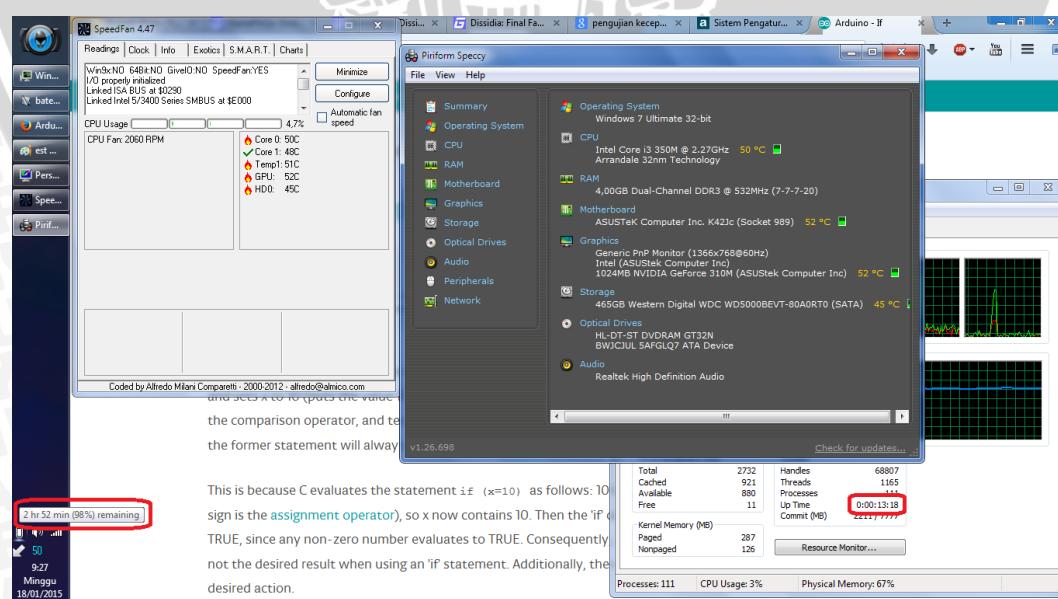
% Error steady state =

$$\begin{aligned}
 &= \frac{1}{N} \cdot \sum_{i=1}^N \frac{|PV(i) - 49|}{49} \cdot 100\% \\
 &= 0,7233 / 49 \cdot 100\% \\
 &= 1,47\%
 \end{aligned}$$

Dan dengan mendapat gangguan berupa penambahan program *browsing* (Mozilla / Chrome), sistem mendapat *error* terbesar sebesar :

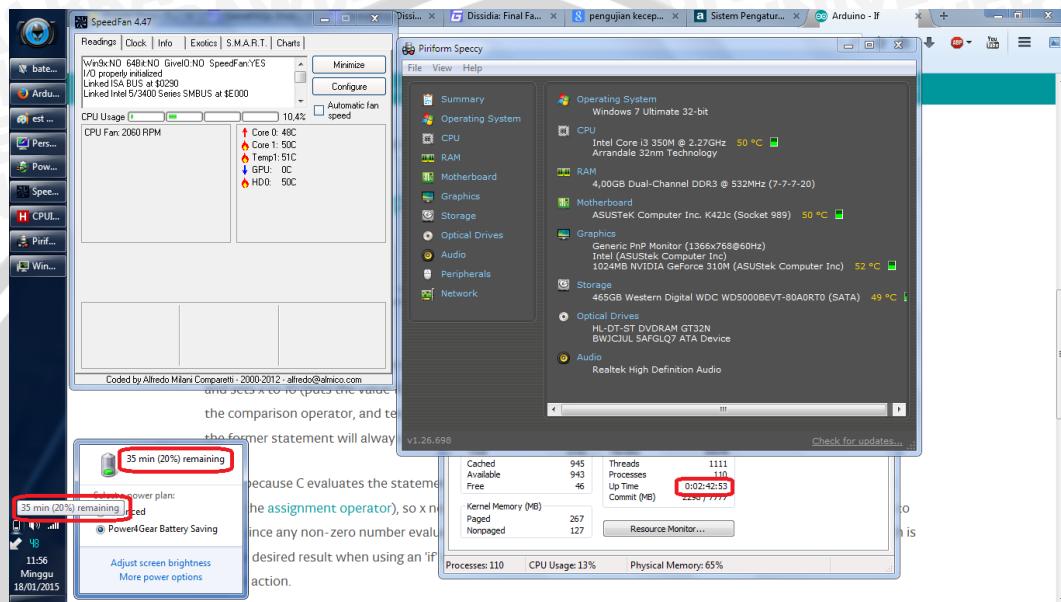
$$\begin{aligned}
 \% \text{ Max error dengan gangguan} &= \frac{|51,41 - 49|}{49} \times 100\% \\
 &= 0,04918 \times 100\% \\
 &= 4,92\%
 \end{aligned}$$

Dari hasil pengujian diatas dapat dilihat respon sistem terhadap gangguan berupa penambahan program Mozilla dan Microsoft Word. Gangguan tersebut menyebabkan perubahan pada temperatur yang menyebabkan terjadi *error*. Secara keseluruhan sistem dapat kembali pada keadaan *steady* dengan *recovery time* sebesar 75 detik dan mampu memberikan respon sistem yang baik ketika terjadinya gangguan. Data pengujian terdapat dalam lampiran. Untuk pengujian waktu pemakaian baterai laptop dapat dilihat dalam Gambar 5.9.



Gambar 5.9 Pemakaian Awal Baterai Laptop dengan Sistem Secara Keseluruhan

Dengan kondisi yang sama pada pengujian sebelumnya, dalam Gambar 5.9 tertera estimasi waktu pemakaian laptop dengan menggunakan baterai selama 2 jam 52 menit. Kipas internal berputar dengan kecepatan rata-rata 2000 RPM. Untuk pemakaian akhir saat baterai mencapai 20 % dapat dilihat dalam Gambar 5.10.



Gambar 5.10 Pemakaian Akhir Baterai Laptop dengan Sistem Secara Keseluruhan

Dalam Gambar 5.9 dan 5.10 dapat disimpulkan bahwa waktu pemakaian aktual baterai mulai dari kapasitas 98 % s/d 20 %, adalah 149 menit atau 2 jam 29 menit.

Dari hasil pengujian Laptop tanpa sistem pengendali dengan pengujian sistem secara keseluruhan, dapat dimimpulkan bahwa sistem mampu mempertahankan baterai laptop lebih lama sebesar 48 menit atau 2880 detik dan baterai berhemat sebesar 47,52 % dengan menggunakan alat ini.

BAB VI PENUTUP

6.1 Kesimpulan

Berdasarkan pengujian dan analisis yang telah dilakukan, dapat ditarik kesimpulan sebagai berikut:

1. Penggunaan KLF sebagai pengendali suhu dengan menggunakan 5 buah label MF, metode inferensi *Min-Max composition*, dan metode defuzzifikasi *Center of Gravity* (COG) dapat mencapai nilai *setpoint* dengan *settling time* (t_s) = 196 detik. Pada pengujian dengan gangguan berupa penambahan program untuk *browsing* (Mozilla, Chrome, dll) mendapatkan *error steady state* sebesar 1,47 %, *error maksimal* sebesar 2,41 °C (4,92 %) dan *recovery time* sebesar 75 detik setelah diberi gangguan.
2. Waktu pemakaian aktual baterai mulai dari kapasitas 98 % s/d 20 %, adalah 149 menit atau 2 jam 29 menit. Kipas internal berputar dengan kecepatan rata-rata 2000 RPM. Sistem mampu mempertahankan baterai laptop lebih lama sebesar 48 menit atau 2880 detik dan baterai berhemat sebesar 47,52 % dengan menggunakan alat ini.

6.2 Saran

Untuk penelitian selanjutnya agar didapatkan hasil yang lebih baik, disarankan untuk :

1. Menggunakan sumber daya baterai agar sistem dapat digunakan secara portabel.
2. Menggunakan aktuator yang lebih cepat agar dapat mengatasi *noise* yang besar (program yang membutuhkan kerja prosesor tinggi).
3. Melakukan pengujian dengan gangguan eksternal, seperti perubahan suhu ruangan akibat perpindahan tempat dan sejenisnya.



DAFTAR PUSTAKA

- Arduino. 2014. Arduino Mega 2560 (Online, <http://arduino.cc/en/Main/ArduinoBoardMega2560>, diakses tanggal 7 November 2014)
- Camelia., Risa Pradita. 2012. *Aplikasi Kontrol Logika Fuzzy untuk Pengaturan Posisi Sudut Putar Motor DC pada Model system Rotary Parking*. Malang: Teknik Elektro Universitas Brawijaya.
- Intel Corporation. 2010. Intel® Core™ i3-350M Processor. (Online, http://ark.intel.com/products/43529/Intel-Core-i3-350M-Processor-3M-Cache-2_26-GHz, diakses tanggal 5 Juni 2014).
- Kuswandi., Son. 2000. *Kendali Cerdas (Intelligent Control)*: EEPIS Press.
- The MathWorks, Inc. 2014. What Is Fuzzy Logic? (Online, <http://www.mathworks.com/help/fuzzy/what-is-fuzzy-logic.html>, diakses tanggal 25 Agustus 2014).
- Texas Instruments. 2013. LM35 Precision Centigrade Temperature Sensors, (Online, <http://www.ti.com>, diakses tanggal 18 Oktober 2014).
- Ogata., Katsuhiko. 1997. *Teknik Kontrol Automatik (Sistem Pengaturan)*. Jakarta: Erlangga.

LAMPIRAN



UNIVERSITAS BRAWIJAYA



LAMPIRAN I**FOTO ALAT DAN DOKUMENTASI**

UNIVERSITAS BRAWIJAYA





Kotak Pengendali Suhu Tampak Samping



Kotak Pengendali Suhu Tampak Depan

LAMPIRAN II**DATA PENGUJIAN**

UNIVERSITAS BRAWIJAYA



Tabel 5.5 Data hasil pengujian sistem

Waktu	Suhu	PWM
0	34,78	95
7	33,43	95
14	34,89	95
21	36,24	95
28	36,13	95
35	37,79	95
42	38,1	95
49	39,04	95
56	38,71	95
63	39,45	95
70	39,66	95
77	39,77	95
84	39,33	95
91	41,01	95
98	40,7	95
105	41,63	95
112	42,49	125
119	43,29	125
126	43,71	125
133	44,12	125
140	44,63	125
147	45,16	125
154	45,47	125
161	45,68	125
168	46,2	125
175	47,13	125

182	47,76	125
189	48,07	125
196	48,81	125
203	48,83	125
210	47,47	125
217	48,79	125
224	48,87	125
231	47,44	125
238	49,42	125
245	49,67	205
252	50,63	125
259	51,41	175
266	51,28	205
273	50,64	175
280	50,36	175
287	50,68	175
294	50,26	175
301	50,73	175
308	50,42	175
315	50,12	175
322	50,23	175
329	49,24	125
336	49,31	175
343	49,64	175
350	48,71	175
357	49,24	125
364	48,83	125

LAMPIRAN III**LISTING PROGRAM**

UNIVERSITAS BRAWIJAYA



// By Fauzan Nusyura - Elektro UB - Fuzzy logic for controling temperature processor

```
#include <LiquidCrystal.h>
#include "Arduino.h"
#include <Wprogram.h>
#define LOW 0
#define MED 1
#define HIGH 2
#define NB 0
#define NS 1
#define Z 2
#define PS 3
#define PB 4
```

```
#define S 95
#define MS 125
#define M 175
#define MF 205
#define F 255
```

```
LiquidCrystal lcd(32, 30, 28, 26, 24, 22);
```

```
//variabel untuk sensor
float totaltempC = 0;
float tempCave = 0;
int tempPin = 0;
float sensor[7];
float maxi = 0,mini = 100;
int C;
int Pros;
float totaltempPros = 0;
float tempProsAve = 0;
```

```
//variabel untuk driver motor
int ENA = 8;
int IN1 = 9;
int IN2 = 10;
```

```
//variabel untuk persamaan garis//
int value;
int result;
int x0;
int x1;
int x2;
int x3;
float trimf(int value,float x0,float x1,float x2);
float grade(int value,float x0,float x1);
float reverse_grade(int value,float x0,float x1);
```

```
//variabel untuk sistem//
```



```

float data_Error;
float last_Error;
float data_delError;
float set_point;

//variabel untuk fuzzy//
int pwm;
int pwm_cek[25];
float Error[5];
float delError[5];
float hasil;
long currentMillis,previousMillis=0;
long interval=10000000;
long X,Y;
boolean set_FlagA=false,set_FlagB=false,set_FlagD=false,set_flagE=false;
float NBe = constrain (data_Error, -16.0, -12.0);
float NSe = constrain (data_Error, -11.0, -7.0);
float Ze = constrain (data_Error, -6.0, 6.0);
float PSe = constrain (data_Error, 7.0, 11.0);
float PBe = constrain (data_Error, 12.0, 16.0);

float NBde = constrain (data_delError, -1.6, -1.2);
float NSde = constrain (data_delError, -1.1, -0.7);
float Zde = constrain (data_delError, -0.6, 0.6);
float PSde = constrain (data_delError, 0.6, 1.1);
float PBde = constrain (data_delError, 1.2, 1.6);

//check rule//
unsigned char rule[5][5]=
{
  {S, S, S, M, M},
  {S, S, S, M, MF},
  {S, S, MS, MF, MF},
  {S, MS, MS, MF, F},
  {MS, MS, M, F, F},
};

void fuzzy();
void check_rule();
void defuzzy();

void setup()
{
  Serial.begin(9600);
  lcd.begin(16, 2);
  pinMode (ENA, OUTPUT);
  pinMode (IN1, OUTPUT);
  pinMode (IN2, OUTPUT);
  Serial.begin (9600);
  digitalWrite (IN1, HIGH);
}

```



}

```
void baca_suhu()
{
{for(C = 0;C<7;C++)
{
    sensor[C] = (5.66 * analogRead(tempPin) * 100.0) / 1024.0;
    sensor[Pros] = (sensor[C] + 19.51) / 1.16;
    lcd.setCursor(0, 0);
    lcd.print("T=");
    lcd.setCursor(0, 1);
    lcd.print("PWM=");
    lcd.setCursor(2, 0);
    lcd.print(sensor[C]);
    //lcd.setCursor(5, 1);
    //lcd.print(sensor[Pros]);
    lcd.setCursor(7, 0);
    lcd.print("/");
    lcd.setCursor(7, 1);
    lcd.print("^");
    totaltempC = totaltempC + sensor[C];
    totaltempPros = totaltempPros + sensor[Pros];
    delay(900);
}}
```

```
tempCave = totaltempC / 7.0;
tempProsAve = totaltempPros / 7.0;
```

```
lcd.setCursor(8, 0);
lcd.print(tempCave);
lcd.setCursor(8, 1);
lcd.print(tempProsAve);
```

```
if(tempCave > maxi) { maxi = tempCave; }
if(tempCave < mini) { mini = tempCave; }
```

```
Serial.println(tempProsAve);
```

```
if(47 < tempProsAve < 51)
{
lcd.setCursor(13, 0);
lcd.print("STA");
lcd.setCursor(13, 1);
lcd.print("BIL");
}
if(tempProsAve < 48)
{
lcd.setCursor(13, 0);
lcd.print("DIN");
lcd.setCursor(13, 1);}
```



```

lcd.print("GIN");
}
if(tempProsAve > 50)
{
lcd.setCursor(13, 0);
lcd.print("PA ");
lcd.setCursor(13, 1);
lcd.print("NAS");
}

totaltempC = 0;
totaltempPros = 0;
}
delay(20);
}

```

```

//bentuk membership function/
float trimf(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)+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;
}

float trapmf(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;
}

// rule fuzzy //

void fuzzy()
{
    Error[NB] = reverse_grade(data_Error,-12,-6);
    Error[NS] = trimf(data_Error,-12,-6,0);
    Error[Z] = trimf(data_Error,-6,0,6);
    Error[PS] = trimf(data_Error,0,6,12);
    Error[PB] = grade(data_Error,6,12);
}

```

```

delError[NB] = reverse_grade(data_delError,-1.2,-0.6);
delError[NS] = trimf(data_delError,-1.2,-0.6,0);
delError[Z] = trimf(data_delError,-0.6,0,0.6);
delError[PS] = trimf(data_delError,0,0.6,1.2);
delError[PB] = grade(data_delError,0.6,1.2);

}

void check_rule()
{
float u_min=0;
float u_max=0;
char x,y;
for(x=0;x<10;x++)
{
    pwm_cek[x]=0;
}
for (x=0;x<5;x++)
{
    for (y=0;y<5;y++)
    {
        if(Error[y]>0&&delError[x])
        {
            u_min=(float)fmin(Error[y],delError[x]);
            pwm_cek[rule[x][y]]=(float)fmax(u_min,pwm_cek[rule[x][y]]);
        }
    }
}
}

// defuzzifikasi //
void defuzzy()
{
if(Error[NB] && delError[NB] || delError[NS] || delError[Z] || delError[PS])
{
pwm = (int) S;
}
if(Error[NS] && delError[NB] || delError[NS] || delError[Z])
{
pwm = (int) S;
}
if(Error[Z] && delError[NB] || delError[NS])
{
pwm = (int) S;
}

if(Error[NB] && delError[PB])
{
pwm = (int) MS;
}
if(Error[NS] && delError[PS] || delError[PB])
{
}
}

```



```

{
pwm = (int) MS;
}
if(Error[Z] && delError[PS] || delError[Z])
{
pwm = (int) MS;
}

if(Error[Z] && delError[PB])
{
pwm = (int) M;
}
if(Error[PS] && delError[NB] || delError[NS])
{
pwm = (int) M;
}
if(Error[PB] && delError[NB])
{
pwm = (int) M;
}

if(Error[PS] && delError[Z] || delError[PS])
{
pwm = (int) MF;
}
if(Error[PB] && delError[Z] || delError[NS])
{
pwm = (int) MF;
}

if(Error[PS] && delError[PB])
{
pwm = (int) F;
}
if(Error[PB] && delError[PB] || delError[PS])
{
pwm = (int) F;
}

analogWrite(ENA,pwm);
lcd.setCursor(4, 1);
lcd.print(pwm);
delay(40);
}

void loop ()
{
baca_suhu();
//sistem//
set_point = 49.0 ;
}

```



```
data_Error = tempProsAve - set_point;  
data_delError = data_Error - last_Error;  
//----- KLF -----//  
fuzzy();  
check_rule();  
defuzzy();  
last_Error = data_Error;  
delay(40);  
}
```



UNIVERSITAS BRAWIJAYA

LAMPIRAN IV**DATASHEET**

UNIVERSITAS BRAWIJAYA

