



**DESAIN SISTEM INKUBATOR BAYI OTOMATIS DENGAN  
METODE KONTROL *FUZZY*-PID**

**SKRIPSI**

Oleh

**Rosida Amalia Nurul Qoyima**

**NIM 161910201029**

**PROGRAM STUDI STRATA 1 TEKNIK ELEKTRO  
JURUSAN TEKNIK ELEKTRO  
FAKULTAS TEKNIK  
UNIVERSITAS JEMBER  
2020**



**DESAIN SISTEM INKUBATOR BAYI OTOMATIS DENGAN  
METODE KONTROL FUZZY-PID**

**SKRIPSI**

diajukan guna melengkapi skripsi dan memenuhi salah satu syarat  
untuk menyelesaikan Program Studi Strata 1 Jurusan Teknik Elektro  
dan mencapai gelar sarjana

Oleh

**Rosida Amalia Nurul Qoyima**

**NIM 161910201029**

**PROGRAM STUDI STRATA 1 TEKNIK ELEKTRO  
JURUSAN TEKNIK ELEKTRO  
FAKULTAS TEKNIK  
UNIVERSITAS JEMBER  
2020**

## PERSEMBAHAN

Alhamdulillah, puji syukur kehadiran Allah SWT atas nikmat dan karuniaNya yang telah melimpahkan rahmat yang tak ternilai sehingga saya bisa menyelesaikan penelitian ini. Banyak doa, semangat dan bantuan yang penulis dapatkan dari berbagai pihak dalam melangsungkan penelitian. Akhirnya, skripsi ini dipersembahkan untuk:

1. Allah SWT, yang Maha Pengasih lagi Maha Penyayang;
2. Nabi besar Muhammad SAW, yang menjadi suri tauladan bagi seluruh umat;
3. Kedua orang tua yang amat saya sayangi, yakni Bapak Muh Nurkoyim Kustanto dan Bunda Umi Muawanah yang selalu membimbing dan mendidik saya. Serta adikku Farisa Fasla Nurul Qoyima dan Firia Renanda Nurlianisa yang selalu menghibur dan memberikan semangat;
4. Dr. Ir. Satryo Budi Utomo, S.T., M.T selaku Dosen Pembimbing Utama dan Arizal Mujibtama Nanda Imron, S.T., M.T selaku Dosen Pembimbing Anggota yang telah membimbing dengan sangat sabar dan memberikan saran-saran demi kesempurnaan skripsi ini;
5. Artika Sari Ramadhani teman satu perjuangan, susah dan senang bersama sejak mahasiswa baru sampai dengan pengerjaan tugas akhir;
6. *Good girl squad* yaitu Nadya, Alif, Dhea, Mayang, Elfrida, Gita, Yuni, Yulia, Uus dan Okhsin yang selalu membantu dan menghibur semasa kuliah;
7. Syaiful, Fikry, Tigo, Cahya, Zaqi, Mas Turasno, Mas Wawan dan seluruh asisten Laboratorium Elektronika Terapan dan teman-teman Elektronika 2016. Terima kasih telah memberikan saran-saran dan solusi dalam diskusi.
8. Ali Aqsho Hudzaifi yang sangat saya andalkan, terima kasih sudah membantu dengan sangat sabar dan ikhlas dalam segala keadaan;
9. Untuk seluruh teman-teman INDUKTRO 2016 yang telah menjadikan kehidupan semasa kuliah menjadi sangat berwarna;

10. Para guru sejak Taman Kanak-kanak hingga Perguruan Tinggi yang terhormat, terima kasih telah memberikan ilmu;
11. Serta rekan-rekan yang penulis kenal dan rekan-rekan yang membaca serta menjadikan skripsi ini sebagai referensi. Terima kasih untuk segalanya.



**MOTTO**

“Barangsiapa belum pernah merasakan pahitnya menuntut ilmu walau sesaat. Ia akan menelan hinanya kebodohan sepanjang hidupnya”

**(Imam Asy-Syafi’i)**

“Merubah diri menjadi lebih baik adalah salah satu tujuan kita dalam hidup ini”

**(Merry Riana)**

“Ingatlah satu hal ketika kamu gagal dan terjatuh; bangkit lagi”

**(Rosida Amalia Nurul Qoyima)**

“Maka sesungguhnya bersama kesulitan, ada kemudahan. Sesungguhnya bersama kesulitan, ada kemudahan”

**(Q.S Al-Insyirah (94): 5-6)**

**PERNYATAAN**

Saya yang bertanda tangan dibawah ini:

Nama : Rosida Amalia Nurul Qoyima

NIM : 161910201029

Menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Desain Sistem Inkubator Bayi Otomatis dengan Metode Kontrol *Fuzzy*-PID” adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada industry manapun dan bukan karya jiplakan. Saya bertanggung jawab penuh atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapatkan sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Jember, 2 Juli 2020

Yang menyatakan,



Rosida Amalia Nurul Qoyima

NIM 161910201029

**SKRIPSI**

**DESAIN SISTEM INKUBATOR BAYI OTOMATIS DENGAN METODE  
KONTROL *FUZZY*-PID**

Oleh

Rosida Amalia Nurul Qoyima

NIM 161910201029

Pembimbing:

Dosen Pembimbing Utama : Dr. Ir. Satrio Budi Utomo, S.T., M.T.

Dosen Pembimbing Anggota : Arizal Mujibtamala Nanda Imron, S.T., M.T.

PENGESAHAN

Skripsi berjudul “Desain Sistem Inkubator Bayi Otomatis dengan Metode Kontrol *Fuzzy-PID*” karya Rosida Amalia Nurul Qoyima telah diuji dan disahkan pada:

Hari : Kamis

Tanggal : 02 Juli 2020

Tempat : Fakultas Teknik Universitas Jember

Tim penguji,

Ketua,



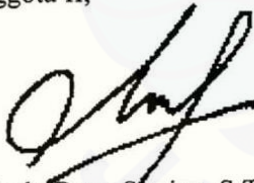
Dr. Ir. Saityo Budi Utomo, S.T., M.T.  
NIP 198501262008011002

Anggota I,



Arizal Mujibtamala Nanda Imron, S.T., M.T.  
NIP 76001717009

Anggota II,



Alfredo Bayu Satriya, S.T., M.T.  
NIP 198905192015041001

Anggota III,



Wahyu Muldayani, S.T., M.T.  
NIP 760016799

Mengesahkan

Dekan,

Dr. Triwahju Hardianto, S.T., M.T.

NIP 1970082619970210011



## RINGKASAN

### **Desain Sistem Inkubator Bayi Otomatis dengan Metode Kontrol *Fuzzy*-PID;**

Rosida Amalia Nurul Qoyima; 161910201029; 2020; 82 Halaman; Jurusan Teknik Elektro Fakultas Teknik Universitas Jember.

Berdasarkan data Biro Pusat Statistik 2016, angka kematian bayi (AKB) mencapai 25 kematian setiap 1.000 bayi. Kemudian pada 1 Januari 2018 menurut PBB yang mengurus masalah anak-anak yaitu UNICEF kelahiran bayi prematur di Indonesia masih menjadi urutan kelima dunia dengan bayi lahir prematur sebanyak 13.370 bayi.

Bayi lahir prematur membutuhkan inkubator untuk menjaga tubuhnya agar tetap hangat, sehingga diperlukan pengontrolan suhu. Permasalahan saat ini adalah harga inkubator import dengan teknologi yang dibutuhkan untuk merawat bayi lahir prematur memiliki harga yang relative mahal yaitu berkisar Rp 16 juta – Rp 118 juta. Namun ada pula inkubator buatan lokal yang memiliki harga lebih terjangkau yaitu berkisar Rp 1 juta – Rp 3 juta tetapi dengan teknologi yang kurang memadai, inkubator lokal dipasaran masih banyak yang dikontrol secara manual. Oleh karena itu tujuan dari penelitian ini adalah merancang inkubator bayi ekonomis namun memiliki sistem kontrol yang otomatis.

Metode yang akan digunakan adalah kontrol *Fuzzy*-PID. Nantinya kestabilan suhu dan kecepatan respon sistem pada inkubator akan diatur menggunakan kontrol *Fuzzy*-PID. Suhu dalam inkubator diukur menggunakan sensor DHT22. Digunakan *error* dan *delta error* dari suhu sebagai parameter *input* dalam perancangan himpunan keanggotaan *fuzzy*. *Heater* yang digunakan dalam penelitian ini adalah lampu pijar agar menghasilkan panas yang aman untuk bayi. Kemudian digunakan *setpoint* untuk mendapatkan suhu yang diinginkan.

Pengujian pertama adalah pengujian kontrol PID dengan *tuning* manual atau *trial and error*. Untuk pengujian keseluruhan alat digunakan logika *fuzzy* sebagai *tuning* PID dengan menjadikan nilai dari hasil *tuning* manual PID sebagai range kondisi pada *rule base* logika *fuzzy*. Kemudian dilakukan pengujian dengan dua analisa, yaitu analisa dengan beban dan analisa tanpa beban. Beban yang

dimaksudkan dalam pengujian adalah air dengan suhu  $36,5^{\circ}\text{C}$ - $37,5^{\circ}\text{C}$ , suhu tersebut menyerupai suhu bayi yang baru lahir.

Pengambilan dan pengumpulan data menggunakan empat *setpoint* yaitu  $32^{\circ}\text{C}$ ,  $33^{\circ}\text{C}$ ,  $34^{\circ}\text{C}$  dan  $35^{\circ}\text{C}$  dimana suhu tersebut adalah suhu yang dibutuhkan bayi pada inkubator menurut aturan berdasarkan berat badan dan usia bayi.

Dalam pengujian kontrol *fuzzy*-PID pada *set point*  $35^{\circ}\text{C}$  menghasilkan *rise time* dan *steady state* yang lebih cepat dibandingkan dengan pengujian kontrol PID. Membutuhkan waktu 207 detik untuk suhu dalam mencapai *setpoint* ketika *setpoint* yang digunakan adalah  $35^{\circ}\text{C}$  dalam pengujian tanpa beban dan dihasilkan *max overshoot* sebesar 0,5% sebagai *overshoot* terbesar dalam seluruh pengujian yang telah dilakukan.



## PRAKATA

Puji syukur kehadirat Allah SWT atas nikmat dan karuniaNya yang telah melimpahkan rahmat yang tak ternilai sehingga saya bisa menyelesaikan penelitian sekaligus penyusunan skripsi yang berjudul "**Desain Sistem Inkubator Bayi Otomatis dengan Metode Kontrol *Fuzzy*-PID**". Skripsi ini disusun untuk memenuhi salah satu syarat dalam penyelesaian pendidikan strata satu (S1) pada Jurusan Teknik Elektro Fakultas Teknik Universitas Jember. Selama penyusunan skripsi ini tentunya terdapat banyak pihak yang telah memberikan bantuan berupa saran, motivasi, fasilitas, doa dan dukungan lainnya. Oleh karena itu penulis menyampaikan terima kasih kepada:

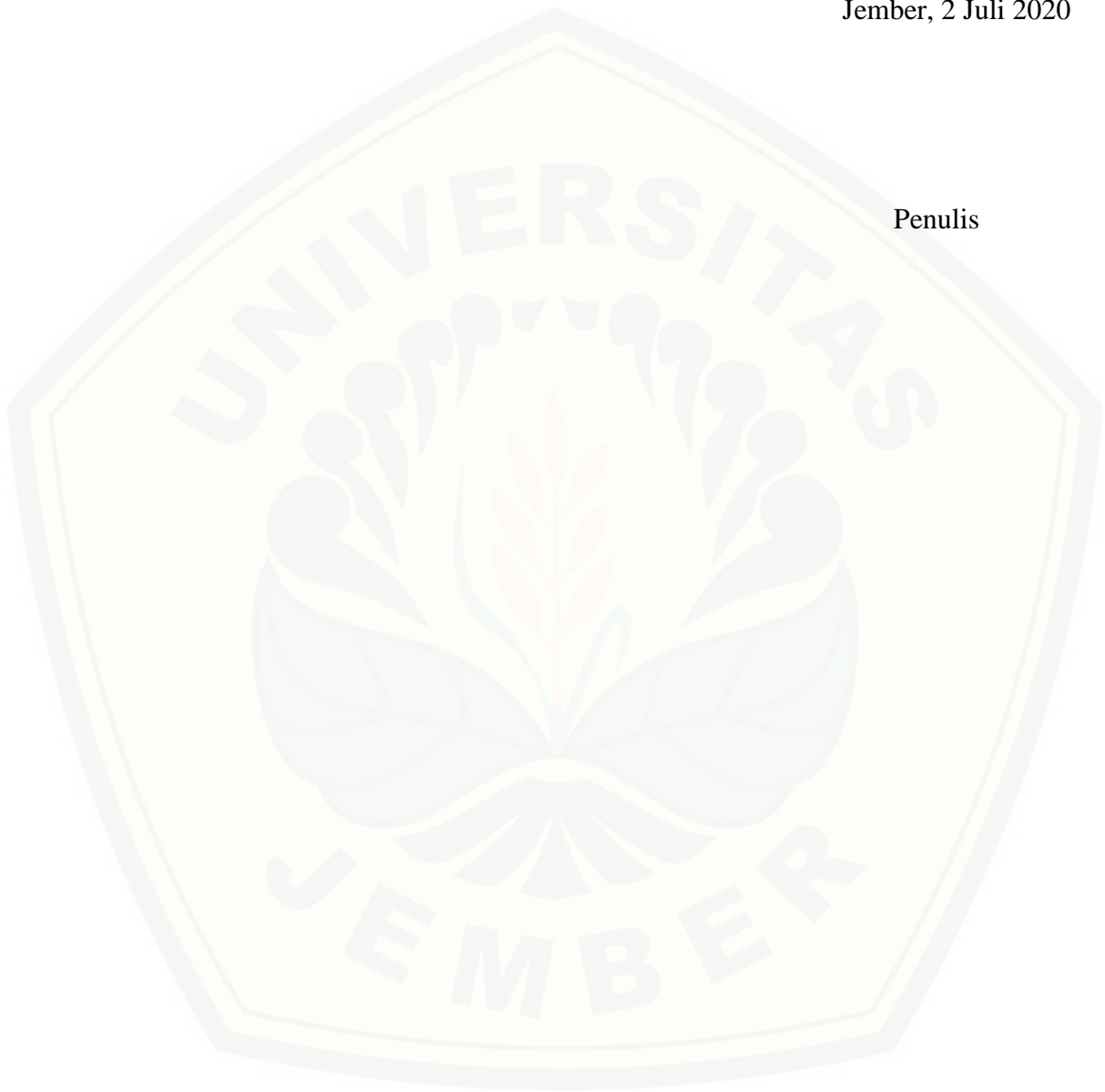
1. Bapak Dr. Triwahju Hardianto, S.T., M.T. selaku Dekan Fakultas Teknik Universitas Jember;
2. Bapak Dr. Bambang Sri Kaloko S.T., M.T. selaku Ketua Jurusan Teknik Elektro Universitas Jember;
3. Dr. Ir. Satrio Budi Utomo, S.T., M.T. selaku Dosen Pembimbing Utama dan Arizal Mujibntama Nanda Imron, S.T., M.T. selaku Dosen Pembimbing Anggota yang telah membimbing dengan sangat sabar dan memberikan saran-saran demi kesempurnaan skripsi ini;
4. Bapak Alfredo Bayu Satriya, S.T., M.T. dan Bapak Wahyu Muldayani, S.T., M.T. selaku dosen penguji yang telah memberikan saran untuk memperbaiki dan menyempurnakan skripsi ini;
5. Keluarga besar Teknik Elektro 2016 (INDUKTRO), terima kasih telah menjadi keluarga yang sangat baik dan luar biasa;
6. Serta seluruh teman-teman seperjuangan yang tidak dapat saya sebutkan satu-satu. Saya sampaikan terima kasih banyak atas semangat, doa dan saran yang telah diberikan.

Penulis menyadari bahwa selama penyusunan skripsi ini masih terdapat banyak kekurangan. Harapan penulis adalah semoga skripsi ini dapat bermanfaat dalam mengembangkan ilmu pengetahuan khususnya dalam bidang teknik elektro. Kritik

dan saran diharapkan dapat terus berlanjut sehingga dapat memperbaiki dan menyempurnakan skripsi ini untuk penelitian selanjutnya.

Jember, 2 Juli 2020

Penulis



DAFTAR ISI

HALAMAN SAMPUL.....	i
HALAMAN JUDUL .....	ii
PERSEMBAHAN.....	iii
MOTTO .....	v
PERNYATAAN.....	vi
HALAMAN PEMBIMBING .....	vii
PENGESAHAN .....	viii
RINGKASAN .....	ix
PRAKATA .....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR.....	xvi
DAFTAR TABEL .....	xvii
DAFTAR LAMPIRAN .....	xix
<b>BAB 1. PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	3
1.3 Batasan Masalah.....	4
1.4 Tujuan Penelitian .....	4
1.5 Manfaat Penelitian.....	4
<b>BAB 2. TINJAUAN PUSTAKA.....</b>	<b>5</b>
2.1 Inkubator Bayi.....	5
2.2 Arduino UNO .....	5
2.3 Sensor DHT22 .....	7
2.4 Modul AC <i>Light Dimmer</i> .....	8
2.5 Modul <i>Driver</i> L298N.....	9
2.6 <i>Fan/Kipas Aksial</i> .....	11
2.7 <i>Heater</i> .....	12
2.8 Aturan Suhu Inkubator Sesuai Berat Badan dan Usia Bayi.....	13
2.9 Kontrol PID ( <i>Proportional Integral Derivative</i> ).....	13
2.10 Kontrol Logika <i>Fuzzy</i> .....	15

2.10.1 Pengertian Logika <i>Fuzzy</i> .....	15
2.10.2 Himpunan <i>Fuzzy</i> .....	15
2.10.3 Fungsi Keanggotaan.....	16
2.10.4 <i>Fuzzyfikasi</i> .....	16
2.10.5 <i>Rule Base</i> .....	16
2.10.6 <i>Defuzzyfication</i> .....	16
<b>BAB 3. METODOLOGI PENELITIAN .....</b>	<b>17</b>
<b>3.1 Waktu dan Tempat Penelitian.....</b>	<b>17</b>
3.1.1 Tempat.....	17
3.1.2 Waktu .....	17
<b>3.2 Tahapan Penelitian.....</b>	<b>18</b>
<b>3.3 Perancangan Perangkat Keras.....</b>	<b>19</b>
3.3.1 Alat dan Bahan .....	19
3.3.2 Desain Alat.....	19
3.3.3 Rancang Bentuk dan Cara Kerja Alat .....	20
3.3.4 Desain Elektrik.....	21
3.3.4.1 Rangkaian Sensor DHT22.....	21
3.3.4.2 Rangkaian <i>Heater</i> dengan Modul <i>AC Light Dimmer</i> .....	22
3.3.4.3 Rangkaian Fan/Kipas Aksial dengan Modul <i>Driver</i> L289N .....	23
<b>3.5 Perancangan Perangkat Lunak.....</b>	<b>24</b>
3.5.1 Sistem Kontrol PID .....	24
3.5.2 Sistem Logika <i>Fuzzy</i> .....	26
<b>3.6 Metode Pengujian dan Pengumpulan Data.....</b>	<b>34</b>
<b>3.7 Metode dan Analisa .....</b>	<b>37</b>
<b>BAB 4. HASIL DAN PEMBAHASAN.....</b>	<b>39</b>
<b>4.1 Pengujian Sensor Suhu dan Kelembaban .....</b>	<b>39</b>
<b>4.2 Pengujian PID dengan <i>Tuning Manual (Trial and Error)</i>.....</b>	<b>43</b>
4.2.1 Pengujian Kontrol P .....	43
4.2.2 Pengujian Kontrol PI.....	44
4.2.3 Pengujian Kontrol PD .....	46
4.2.4 Pengujian Kontrol PID.....	47

4.3 Pengujian Kontrol <i>Fuzzy</i> -PID .....	48
<b>BAB 5. PENUTUP</b> .....	<b>55</b>
5.1 Kesimpulan.....	55
5.2 Saran .....	55
<b>DAFTAR PUSTAKA</b> .....	<b>56</b>
<b>LAMPIRAN</b> .....	<b>58</b>



DAFTAR GAMBAR

Gambar 2.1 Inkubator Bayi.....	5
Gambar 2.2 Arduino UNO.....	6
Gambar 2.3 Sensor DHT22.....	7
Gambar 2.4 Modul <i>AC Light Dimmer</i> .....	9
Gambar 2.5 Modul <i>Driver L298N</i> .....	9
Gambar 2.6 <i>Fan</i> .....	11
Gambar 2.7 Lampu Pijar .....	12
Gambar 2.8 Blok Diagram Kontrol PID .....	14
Gambar 2.9 Blok Diagram Logika <i>Fuzzy</i> .....	15
Gambar 3.1 Desain Mekanik Inkubator Bayi .....	19
Gambar 3.2 (a) Inkubator Bayi, (b) Rangkaian Kontroler .....	20
Gambar 3.3 Diagram Blok Inkubator Bayi .....	20
Gambar 3.4 Rangkaian DHT22 dengan Arduino.....	21
Gambar 3.5 Rangkaian <i>Heater</i> dengan Modul <i>AC Light Dimmer</i> dan Arduino Uno.....	23
Gambar 3.6 Rangkaian Kipas Aksial dengan Modul <i>Driver L298N</i> dan 25 Arduino Uno.....	24
Gambar 3.7 <i>Flowchart</i> Program Kontrol PID.....	25
Gambar 3.8 <i>Membership Function Input Error Suhu</i> .....	27
Gambar 3.9 <i>Membership Function Input Delta Error Suhu</i> .....	28
Gambar 3.10 <i>Membership Function Output Kp</i> .....	28
Gambar 3.11 <i>Membership Function Output Ki</i> .....	29
Gambar 3.12 <i>Membership Function Output Kd</i> .....	29
Gambar 3.13 <i>Flowchart</i> Kontrol <i>Fuzzy-PID</i> .....	33
Gambar 3.14 Pengujian Sensor .....	34
Gambar 3.15 Pengujian Kontrol P .....	35
Gambar 3.16 Pengujian Kontrol PI.....	35
Gambar 3.17 Pengujian Kontrol PD .....	36
Gambar 3.18 Pengujian Kontrol PID .....	36
Gambar 3.19 Pengujian Kontrol <i>Fuzzy-PID</i> .....	37



Gambar 3.20 Pengujian Tanpa Beban dengan Kontrol <i>Fuzzy</i> -PID.....	37
Gambar 3.21 Pengujian dengan Beban dengan Kontrol <i>Fuzzy</i> -PID .....	37
Gambar 4.1 Grafik Kelinearan Hasil Pengujian Sensor Suhu .....	41
Gambar 4.2 Grafik Kelinearan Hasil Pengujian Sensor Kelembaban .....	41
Gambar 4.3 Grafik Respon Keluaran Kontrol P .....	43
Gambar 4.4 Grafik Respon Keluaran Kontrol PI.....	45
Gambar 4.5 Grafik Respon Keluaran Kontrol PD .....	46
Gambar 4.6 Grafik Respon Keluaran Kontrol PID .....	47
Gambar 4.7 Grafik Respon Kontrol <i>Fuzzy</i> -PID.....	48
Gambar 4.8 (a) Grafik Respon Kontrol Pengujian Tanpa Beban Pada <i>Set point</i> 32°C, (b) Grafik Respon Kontrol Pengujian Dengan Beban Pada <i>Set point</i> 32°C.....	49
Gambar 4.9 (a) Grafik Respon Kontrol Pengujian Tanpa Beban Pada <i>Setpoint</i> 33°C, (b) Grafik Respon Kontrol Pengujian Dengan Beban Pada <i>Setpoint</i> 33°C.....	50
Gambar 4.10 (a) Grafik Respon Kontrol Pengujian Tanpa Beban Pada <i>Setpoint</i> 34°C, (b) Grafik Respon Kontrol Pengujian Dengan Beban Pada <i>Setpoint</i> 34°C .....	51
Gambar 4.11 (a) Grafik Respon Kontrol Pengujian Tanpa Beban Pada <i>Setpoint</i> 35°C, (b) Grafik Respon Kontrol Pengujian Dengan Beban Pada <i>Setpoint</i> 35°C .....	52

DAFTAR TABEL

Tabel 2.1 Spesifikasi Arduino UNO .....	7
Tabel 2.2 Spesifikasi Sensor DHT22 .....	8
Tabel 2.3 Spesifikasi dari Modul <i>Driver</i> L298N .....	11
Tabel 2.4 Aturan Suhu Udara Inkubator Sesuai Berat Badan dan Usia Bayi .....	13
Tabel 3.1 <i>Timeline</i> Penelitian.....	17
Tabel 3.2 <i>Rule Output</i> Kp Suhu .....	30
Tabel 3.3 <i>Rule Output</i> Ki Suhu .....	30
Tabel 3.4 <i>Rule Output</i> Kd Suhu .....	31
Tabel 4.1 Perbandingan Sensor DHT22 dengan Digital Termometer dan Higrometer .....	40
Tabel 4.2 Perbandingan Hasil Kalibrasi Sensor DHT22 dengan Digital Termometer dan Higrometer .....	42
Tabel 4.3 Tanggapan Suhu dengan Kontrol P .....	44
Tabel 4.4 Tanggapan Suhu dengan Kontrol PI .....	45
Tabel 4.5 Tanggapan Suhu dengan Kontrol PD .....	46
Tabel 4.6 Tanggapan Suhu dengan Kontrol PID .....	48
Tabel 4.7 Tanggapan Suhu dengan Kontrol <i>Fuzzy</i> -PID.....	49
Tabel 4.8 Tanggapan Suhu pada <i>Setpoint</i> 32°C .....	50
Tabel 4.9 Tanggapan Suhu pada <i>Setpoint</i> 33°C .....	50
Tabel 4.10 Tanggapan Suhu pada <i>Setpoint</i> 34°C .....	51
Tabel 4.11 Tanggapan Suhu pada <i>Setpoint</i> 35°C .....	52

**DAFTAR LAMPIRAN**

1. <i>Listing</i> Program Arduino Uno.....	58
2. Gambar <i>Hardware</i> .....	70
3. Biaya Pengeluaran Pembuatan Inkubator Bayi.....	73



## BAB 1. PENDAHULUAN

### 1.1 Latar Belakang

Kelahiran prematur adalah kelahiran yang terjadi lebih awal dari hari perkiraan lahir atau usia kandungan belum 37 minggu. Kondisi ini terjadi ketika kontraksi rahim mengakibatkan terbukanya leher rahim (serviks), sehingga membuat janin memasuki jalan lahir. Minggu terakhir masa kehamilan merupakan masa yang penting dalam pembentukan tahap akhir berbagai organ vital, termasuk otak dan paru-paru, serta proses peningkatan berat badan janin. Ada banyak bukti yang menunjukkan bahwa bayi yang lahir prematur memiliki pertumbuhan pascakelahiran yang buruk, dan semakin dini bayi itu, semakin besar gangguan yang mungkin terjadi dan semakin lama ia akan bertahan. Oleh karena itu, bayi yang lahir prematur membutuhkan perawatan intensif, perawatan dapat dilakukan dengan cara *mother kangaroo care* dan menggunakan inkubator.

Di Indonesia diketahui jumlah kelahiran bayi prematur sangat tinggi yaitu mencapai urutan kelima dunia dengan jumlah 675.700 bayi pada tahun 2010 menurut laporan *Born too Soon* milik The Global Action Report on Preterm Birth dari PBB. Berdasarkan data Biro Pusat Statistik 2016, angka kematian bayi (AKB) mencapai 25 kematian setiap 1.000 bayi. Kemudian pada 1 Januari 2018 menurut PBB yang mengurus masalah anak-anak yaitu UNICEF kelahiran bayi prematur di Indonesia masih menjadi urutan kelima dunia dengan bayi lahir prematur sebanyak 13.370 bayi.

Bayi lahir prematur membutuhkan inkubator untuk menjaga tubuhnya agar tetap hangat, sehingga diperlukan pengontrolan suhu. Inkubator digunakan untuk meningkatkan kelangsungan hidup bayi dengan menyediakan lingkungan yang hangat. Permasalahan saat ini adalah harga inkubator import dengan teknologi yang dibutuhkan untuk merawat bayi lahir prematur memiliki harga yang relative mahal yaitu berkisar Rp 16 juta – Rp 118 juta. Namun ada pula inkubator buatan lokal yang memiliki harga lebih terjangkau yaitu berkisar Rp 1 juta – Rp 3 juta tetapi dengan teknologi yang kurang memadai, inkubator lokal dipasaran masih banyak yang dikontrol secara manual. Oleh karena itu tujuan dari penelitian ini adalah merancang inkubator bayi ekonomis namun memiliki sistem kontrol yang otomatis.

Berdasarkan penelitian - penelitian sebelumnya, saya ingin mengembangkan inkubator bayi dengan menggunakan metode yang berbeda agar dapat menghasilkan respon sistem yang lebih baik dan aman untuk bayi dengan pengontrolan kestabilan suhu. Metode yang akan digunakan adalah kontrol *Fuzzy-PID*. Kontrol *Fuzzy-PID* merupakan perpaduan PID konvensional dengan logika *fuzzy* yang mempertahankan struktur linier yang sama dari bagian proporsional, integral, dan turunan tetapi memiliki koefisien konstan serta keuntungan kontrol yang diatur sendiri, pengontrolan ini menyajikan tanggapan yang lebih baik dan efektif. Kontrol PID merupakan pengontrolan paling populer yang digunakan dalam industri karena efektivitasnya yang luar biasa, kesederhanaan implementasi dan penerapan yang luas, namun tidak mudah untuk menentukan nilai pengaturan yang baik dan sesuai. Logika *fuzzy* menambahkan penalaran manusia kedalam inferensial mekanismenya. Nantinya kestabilan suhu dan kecepatan respon sistem pada inkubator akan diatur menggunakan kontrol *Fuzzy-PID*. Dengan kombinasi dari kontroler PID dan kontroler *Fuzzy* dapat meningkatkan lebih lanjut kinerja respon transien dan *steady state*. Parameter pada PID disesuaikan oleh algoritma penalaran *fuzzy*, sehingga memiliki kemampuan *self-adapting*. (Dequan dkk, 2012)

Kesulitan dalam menggunakan kontrol logika *fuzzy* adalah dalam merancang *membership function* dengan benar dan *rule base*. Kelebihan logika *fuzzy* ada pada kemampuan penalaran secara bahasa. Sehingga, dalam perancangannya tidak memerlukan persamaan matematis yang kompleks dari objek yang akan dikendalikan. (Wahyumithali Hameliani, 2017)

Untuk mendesain sistem *fuzzy-PID* hal pertama yang harus dilakukan adalah memilih tipe kontroler *fuzzy-PID* dari *fuzzy-PI*, PD, *fuzzy PI+D* atau *fuzzy PD+I*. Jenis kontrol *fuzzy* harus cocok dengan *plan* yang dikendalikan. Kontroler *fuzzy-PID* dapat dinyatakan sebagai pengontrolan mode geser yang memiliki dua istilah nonlinier, yang pertama bertindak sebagai *equivalent control* dan yang lain bertindak sebagai *switching control*. (Duan dkk. 2012)

Hasil penggunaan sistem kendali logika *fuzzy* untuk pengendalian suhu pada *plant* inkubator bayi adalah kesalahan yang terjadi dapat dikurangi dan kestabilan dapat dipertahankan. Meskipun adanya gangguan yang diberikan pada

sistem, dengan pemanfaatan sistem kendali *fuzzy* ini, dapat menjaga sistem pada keadaan yang stabil. (Setyaningsih dan Murti, 2016)

Dalam perancangan sistem inkubator ini nantinya kontroler *fuzzy* akan diadaptasi oleh PID sebagai tuning. Digunakan *error* dan *delta error* dari suhu sebagai parameter *input* dalam perancangan himpunan keanggotaan *fuzzy*. Dalam perancangan logika *fuzzy* dilakukan dengan beberapa tahap yaitu *fuzzification* dengan melakukan pembentukan *membership function*, kemudian *inference* dengan pembentukan *rule base* dan yang terakhir adalah *defuzzification*. Output dari kontroler *fuzzy* berasal dari *fuzzifikasi* input dan output menggunakan *membership function*. Suhu dan kelembaban pada inkubator diukur dengan menggunakan sensor DHT22. Kemudian menggunakan AC *light dimmer* untuk mengatur kecerahan lampu dan *driver* l298n untuk mengatur kipas. Untuk mikrokontroler yang digunakan adalah arduino uno. Terdapat dua analisis dalam pengujian pada penelitian ini, yaitu analisis tanpa beban dan analisis dengan beban. Beban yang digunakan adalah air dengan suhu 36,5°C-37,5°C, suhu tersebut menyerupai suhu bayi baru lahir.

Pada inkubator ini lampu digunakan sebagai *heater*. Bahan *heater* memberikan pengaruh terhadap cepat tidaknya pemanasan ruangan dan keamanan bagi bayi. Dari beberapa jenis *heater* dipilih lampu karena lampu dapat memberikan panas yang merata pada ruangan serta aman digunakan (Budiono, 2013).

## 1.2 Rumusan Masalah

Adapun beberapa rumusan masalah yang dibuat dalam melakukan penelitian ini sebagai berikut:

1. Bagaimana mendesain inkubator bayi prematur yang memiliki sistem kontrol otomatis dengan respon keluaran yang cepat?
2. Bagaimana cara merancang inkubator bayi prematur yang stabil dan aman untuk bayi lahir prematur?

### 1.3 Batasan Masalah

Adapun beberapa batasan masalah yang dibuat dalam melakukan penelitian ini sebagai berikut:

1. Penelitian ini hanya menggunakan metode *Fuzzy-PID controller* sebagai algoritmanya.
2. Penelitian ini hanya melakukan pengontrolan dengan metode *Fuzzy-PID* untuk suhu.
3. Penelitian ini hanya menggunakan mikrokontroler arduino uno.
4. Sensor yang digunakan hanya DHT22.
5. Ketersediaan sumber tegangan dari PLN tidak terputus atau padam.
6. Tidak menganalisa kebutuhan daya inkubator bayi.
7. Analisis dititik beratkan pada pengontrolan suhu otomatis.

### 1.4 Tujuan Penelitian

Adapun beberapa tujuan penelitian yang dibuat adalah sebagai berikut:

1. Membuat inkubator bayi prematur yang memiliki sistem kontrol otomatis untuk menjaga bayi agar tetap hangat serta memiliki respon sistem yang cepat.
2. Menghasilkan respon sistem dengan *overshoot* yang kecil dan kestabilan suhu menggunakan kontrol *Fuzzy-PID*.

### 1.5 Manfaat Penelitian

Adapun beberapa manfaat yang dibuat dalam melakukan penelitian ini yaitu sebagai berikut:

1. Membantu menjaga keselamatan bayi lahir prematur dengan kontrol suhu otomatis.
2. Mengetahui pengaruh Kontrol *Fuzzy-PID* pada kestabilan kontrol suhu dan kecepatan sistem inkubator bayi prematur.

## BAB 2. TINJAUAN PUSTAKA

### 2.1 Inkubator Bayi

Inkubator bayi adalah *box* yang digunakan untuk merawat bayi lahir prematur. Fungsi utama dari inkubator adalah menjaga agar tubuh bayi tetap hangat. *Box* inkubator memiliki ukuran tertentu dan dilengkapi alat-alat medis untuk membantu perawatan bayi lahir prematur. Pada inkubator terdapat dua bagian kontrol, yaitu pada bagian atas dan bagian bawah. Pada *box* bagian atas diletakkan sensor dan beberapa rangkaian elektronik medis lainnya, sedangkan pada *box* bagian bawah memiliki tiga bagian utama yaitu *heater*, mikrokontroler dan fan/kipas aksial. Kedua bagian tersebut berperan penting untuk kestabilan suhu dan kelembaban pada inkubator. Pada Gambar 2.1 inkubator juga memiliki fungsi sebagai *monitoring* keadaan bayi.



Gambar 2.1 Inkubator Bayi

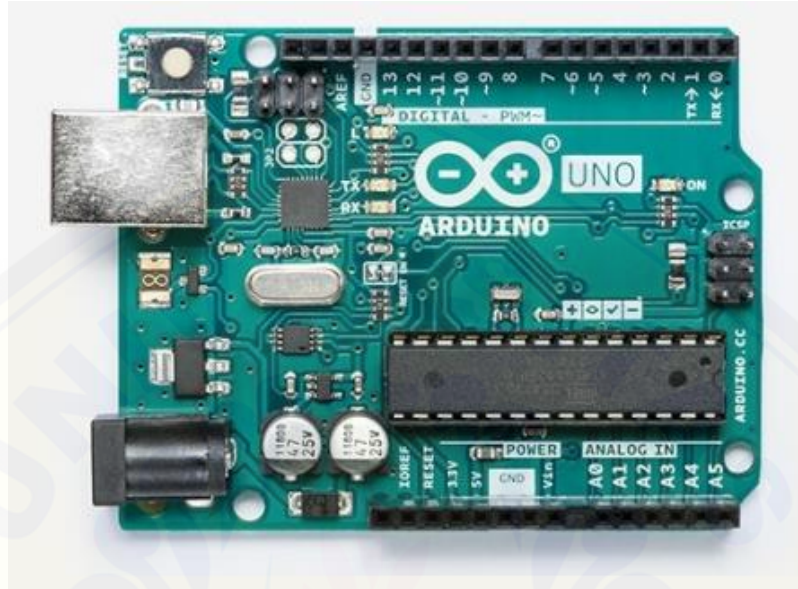
(Sumber: <https://www.bukalapak.com>)

### 2.2 Arduino UNO

Arduino adalah platform pembuatan prototipe elektronik yang bersifat opensource hardware yang berdasarkan pada perangkat keras dan perangkat lunak yang fleksibel dan mudah digunakan. Arduino ditujukan bagi para seniman, desainer, dan siapapun yang tertarik dalam menciptakan objek atau lingkungan yang interaktif. Arduino pada awalnya dikembangkan di Ivrea, Italia. Nama Arduino adalah sebuah nama maskulin yang berarti teman yang kuat. Platform arduino terdiri dari arduino board, shield, bahasa pemrograman arduino, dan arduino development environment. Arduino board biasanya memiliki sebuah chip



dasar mikrokontroler Atmel AVR ATmega8 berikut turunannya. Bahasa pemrograman arduino adalah bahasa pemrograman yang umum digunakan untuk membuat perangkat lunak yang ditanamkan pada arduino board. Bahasa pemrograman arduino mirip dengan bahasa pemrograman C++.



Gambar 2.2 Arduino UNO  
(Sumber: [www.arduino.cc](http://www.arduino.cc))

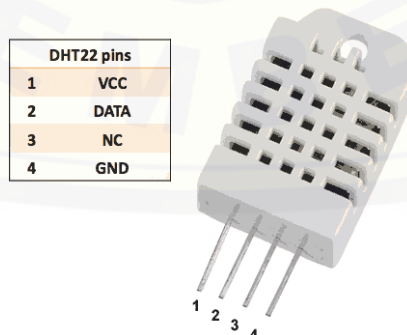
Arduino Uno pada Gambar 2.2 adalah arduino board yang menggunakan mikrokontroler ATmega328. Pada Tabel 2.1 dijelaskan bahwa arduino uno memiliki 14 pin digital (6 pin dapat digunakan sebagai output PWM), 6 input analog, sebuah 16 MHz osilator kristal, sebuah koneksi USB, sebuah konektor sumber tegangan, sebuah header ICSP, dan sebuah tombol reset. Arduino Uno memiliki area cakupan yang luas untuk segala hal yang dibutuhkan untuk mendukung sebuah aplikasi yang berbasis mikrokontroler. Hanya dengan menghubungkannya ke sebuah komputer melalui USB atau memberikan tegangan DC dari baterai atau adaptor AC ke DC sudah dapat membuat aplikasinya bekerja. Arduino Uno menggunakan ATmega16U2 yang diprogram sebagai USB-to-serial converter untuk komunikasi serial ke komputer melalui port USB.

Tabel 2.1 Spesifikasi Arduino UNO

<b>Mikrokontroler</b>	<b>ATmega328</b>
<b>Tegangan Operasi</b>	5V
<b>Tegangan Input (Rekomendasi)</b>	7 – 12V
<b>Tegangan Input (Limit)</b>	6 – 20V
<b>Jumlah Pin I/O Digital</b>	14 (6 diantaranya menyediakan keluaran PWM)
<b>Jumlah Pin Input Analog</b>	6
<b>Arus DC per I/O</b>	40 mA
<b>Arus DC untuk pin 3.3V</b>	50 mA
<b>Flash Memory</b>	32KB dengan 0,5 digunakan sebagai bootloader
<b>SRAM</b>	2KB
<b>EEPROM</b>	1KB
<b>Clock Speed</b>	16MHz

### 2.3 Sensor DHT22

Modul DHT22 merupakan sensor gabungan dari sensor suhu dan sensor kelembaban. Sensor DHT22 memiliki output digital yang sudah dikalibrasi. Sensor ini memiliki kalibrasi akurat dengan kompensasi suhu ruang penyesuaian dengan nilai koefisien tersimpan dalam memori OTP terpadu. Sensor DHT22 memiliki rentang pengukuran suhu dan kelembaban yang luas. Dalam modul DHT22 terdapat komponen sensor suhu NTC (*thermistor*) dan IC disisi belakang sensor.



Gambar 2.3 Sensor DHT22

(Sumber: <https://www.electroschematics.com>)

Sensor DHT22 seperti pada Gambar 2.3 dengan jenis AM2302 memiliki sampling 0,5 Hz atau satu pembacaan setiap dua detik. Tegangan operasinya adalah 3 sampai 5 volt dan arus maksimum yang digunakan adalah 2,5 mA. *Range* pengukuran sensor DHT22 adalah  $-40^{\circ}\text{C}$  sampai  $80^{\circ}\text{C}$  dengan tingkat akurasi  $\pm 0,5^{\circ}\text{C}$  kemudian untuk suhu 0 sampai 100% dengan tingkat akurasi  $\pm 5^{\circ}\text{C}$ . Spesifikasi secara lebih lengkap dijelaskan pada Tabel 2.2, salah satunya adalah sensor DHT memiliki tingkat akurasi  $\pm 2\%$ ; untuk kelembaban dan  $\pm 0,2^{\circ}$  untuk suhu.

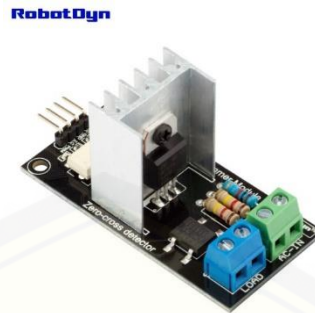
Tabel 2.2 Spesifikasi Sensor DHT22

<b>Power Supply</b>	3,3 V – 6 V DC
<b>Output Signal</b>	Single bus (digital)
<b>Sensing Element</b>	Polymer Humidity Capacitor & DS18B20
<b>Measuring Range</b>	Humidity 0 – 100%RH; Temperature $-40^{\circ}\text{C}$ – $+80^{\circ}\text{C}$
<b>Resolution or Sensitivity</b>	Humidity 0,1%RH; Temperature 0,1 $^{\circ}\text{C}$
<b>Accuracy</b>	Humidity $\pm 2\%$ ; Temperature $\pm 0,2^{\circ}$
<b>Sensing Period</b>	Average: 2s
<b>Dimensions</b>	Small size 14x18x5,5mm; big size 22x28x5mm

#### 2.4 Modul AC *Light Dimmer*

Dimmer merupakan rangkaian elektronik yang memodifikasi bentuk sinyal ac murni menjadi sinyal terpotong-potong sehingga daya keluaran bisa diatur. Pemotongan sinyal ac ini berguna sebagai peredup lampu, memperlambat motor, mengatur pemanasan dan lainnya. Dengan adanya fitur pin zero crossing detector di modul ini, membuat *microcontroller* dapat mengetahui timing yang tepat untuk mengirim sinyal PWM. Tanpa timing yang tepat, arus AC dengan TRIAC jika gatenya dicontrol PWM malah akan kacau sinyal outputnya, menyebabkan dimmer tidak berfungsi dalam menghasilkan sinyal PSM dengan kurva yang benar. Modul

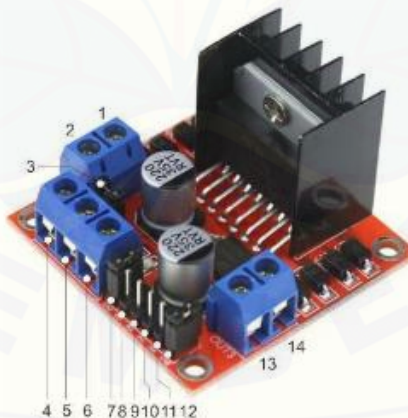
*AC light dimmer* pada Gambar 2.4 dapat dikontrol oleh mikrokontroler, seperti arduino dan raspberry pi.



Gambar 2.4 Modul *AC Light Dimmer*  
(Sumber: <https://robotdyn.com>)

### 2.5 Modul *Driver L298N*

Modul *Driver L298N* merupakan module driver motor DC yang paling banyak digunakan atau dipakai di dunia elektronika yang difungsikan untuk mengontrol kecepatan serta arah perputaran motor DC. Setiap Pin dari modul *driver L298N* seperti pada Gambar 2.5 mempunyai fungsi yang berbeda.



Gambar 2.5 Modul *Driver L298N*  
(Sumber: Budiyanata, dkk. 2018)

Berikut adalah keterangan tiap – tiap pin konektor dari modul *driver* L298N:

1. Kutub (+) dari motor DC 1
2. Kutub (-) dari motor DC 1
3. Jumper untuk tegangan suplai 12V
4. Konektor tegangan masuka
5. *Ground*
6. Konektor tegangan keluaran 5V
7. Jumper untuk Motor DC 1
8. Input data 1 dari mikrokontroler
9. Input data 2 dari mikrokontroler
10. Input data 3 dari mikrokontroler
11. Input data 4 dari mikrokontroler
12. *Jumper* untuk motor DC 2
13. Kutub (+) dari motor DC2
14. Kutub (-) dari motor DC2

IC L298 merupakan sebuah IC tipe H-bridge yang mampu mengendalikan beban-beban induktif seperti relay, solenoid, motor DC dan motor stepper. Pada IC L298 terdiri dari transistor-transistor logik (TTL) dengan gerbang nand yang berfungsi untuk memudahkan dalam menentukan arah putaran suatu motor dc maupun motor stepper. Untuk dipasaran sudah terdapat modul driver motor menggunakan ic l298 ini, sehingga lebih praktis dalam penggunaannya karena pin I/O nya sudah terpackage dengan rapi dan mudah digunakan.

Kelebihan akan modul driver motor L298N ini yaitu dalam hal kepresisian dalam mengontrol motor sehingga motor lebih mudah untuk dikontrol. Untuk spesifikasi dari modul driver l298n dijelaskan pada Tabel 2.3, salah satunya adalah menggunakan tegangan operasi sebesar 5V.

Tabel 2.3 Spesifikasi dari Modul *Driver* L298N

<b>Type</b>	<b>Dual H-Bridge</b>
<b>Chip Kontrol</b>	S T L298N
<b>Tegangan Operasi</b>	5V
<b>Tegangan Minimal Masukan Power</b>	5 – 35V
<b>Arus Input</b>	0 – 36mA
<b>Arus Maksimal Keluaran Per Output</b>	2A
<b>Daya Maksimal</b>	25W

## 2.6 *Fan*/Kipas Aksial

*Axial Fan* adalah sebuah alat yang berfungsi menghasilkan aliran fluida gas searah dengan poros kerja kipas tersebut. Kipas Aksial adalah kipas yang banyak digunakan dalam kehidupan sehari-hari dan paling banyak digunakan karena kemudahan desain serta harga yang lebih ekonomis daripada kipas jenis lain. Kipas aksial mempunyai desain yang tidak terlalu rumit serta dapat menghasilkan *flow* yang besar. (Fathor Rohman, 2017:14).

Gambar 2.6 *Fan*

(Sumber: <https://www.bhinneka.com>)

Kipas Aksial pada Gambar 2.6 biasanya digunakan sebagai pendingin CPU, mesin kendaraan bermotor dan juga banyak digunakan pada dunia industri. Salah satunya digunakan pada pembangkit listrik tenaga uap sebagai *Secondary Air Fan*. Kipas ini berfungsi untuk men-*supply* udara dalam jumlah banyak yang dibutuhkan untuk proses pembakaran pada *furnace boiler*. Kipas ini memiliki dua tingkat

(*stage*) impeller, yang kedua-duanya dapat diatur besar bukaan *pitch*-nya. Hal tersebut berfungsi untuk mengatur jumlah (debit) udara yang akan dikirim untuk proses pembakaran.

### 2.7 Heater

*Heater* terdapat 3 jenis golongan berdasarkan sumber energi primernya, heater listrik, gas, dan minyak tanah. *Heater* listrik terbagi pula menjadi 2 bagian yaitu sistem langsung (*direct*) dan sistem tak langsung (*indirect*). Maksud dari sistem langsung adalah terjadi konversi energi dari energi listrik menjadi panas tanpa moda perantara, dapat dijumpai pada jenis heater konveksi (*electric fan heater*) dan radiasi (*lamp heater*). Sedangkan pada sistem tak langsung terdapat pada heat pump (kebalikan dari fungsi AC), heater gas (*gas heater*) dan minyak tanah (*oil heater*). Pada penelitian ini menggunakan heater listrik jenis radiasi (*lamp heater*) berupa lampu pijar. Heater listrik jenis ini menggunakan prinsip radiasi. Yaitu perpindahan energi melalui gelombang elektromagnetik. Apabila sebuah permukaan terkena oleh radiasi panas tersebut, maka suhu (*temperature*) akan naik. Dengan demikian, heater pada Gambar 2.7 ini akan berfungsi memanaskan benda-benda yang dikenai sinar dari lampu heater. Dikarenakan partikel udara memiliki kerapatan yang rendah sehingga sebagian radiasi panas yang berhasil ditangkap oleh partikel udara.



Gambar 2.7 Lampu Pijar  
(Sumber: Bambang Priyandono, 2013)

## 2.8 Aturan Suhu Inkubator Sesuai Berat Badan dan Usia Bayi

Bayi prematur memiliki usia dan berat badan yang berbeda-beda, sehingga dibutuhkan aturan suhu pada inkubator yang sesuai berdasarkan dari berat badan dan usia bayi. Aturan suhu inkubator dapat dilihat pada Tabel 2.4.

Tabel 2.4 Aturan Suhu Udara Inkubator Sesuai Berat Badan dan Usia Bayi

Berat Badan Bayi	Usia Bayi	Suhu Udara Inkubator
<b>&lt; 1500 gram</b>	1 – 10 hari	35°C
	11 – 3 minggu	34°C
	3 – 5 minggu	33°C
	> 5 minggu	32°C
<b>1500 – 2000 gram</b>	1 – 10 hari	34°C
	11 hari – 4 minggu	33°C
	> 4 minggu	32°C
<b>2100 – 2500 gram</b>	1 – 2 hari	34°C
	3 hari – 3 minggu	33°C
	> 3 minggu	32°C
<b>&gt; 2500 gram</b>	1 – 2 hari	33°C
	> 2 hari	32°C

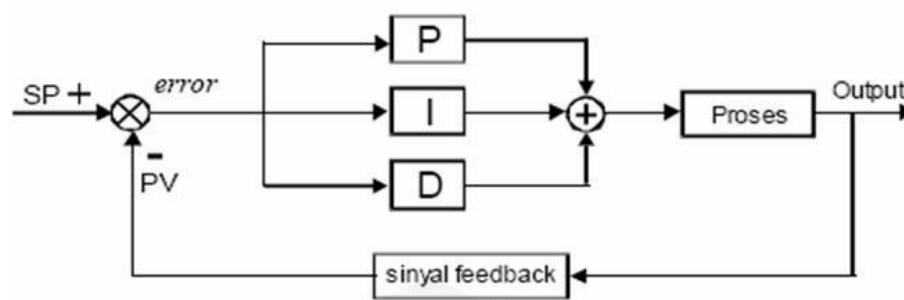
(Sumber: Pratiwi, dkk. 2014)

Dari Tabel 2.4 nantinya akan digunakan sebagian suhu set point setelah diketahui usia dan berat badan dari bayi prematur.

## 2.9 Kontrol PID (*Proportional Integral Derivative*)

PID (Proportional–Integral–Derivative controller) merupakan kontroler untuk menentukan presisi suatu sistem instrumentasi dengan karakteristik adanya umpan balik pada sistem tersebut. Pengontrol PID adalah pengontrol konvensional yang banyak dipakai dalam dunia industri. Pengontrol PID seperti gambar 2.8 akan memberikan aksi kepada Control Valve berdasarkan besar error yang diperoleh. Control valve akan menjadi aktuator yang mengatur aliran fluida dalam proses industri yang terjadi Level air yang diinginkan disebut dengan Set Point. Error adalah perbedaan dari Set Point dengan level air aktual.





Gambar 2.8 Blok Diagram Kontrol PID

(Sumber: Fathor Rohman, 2017)

Adapun persamaan Pengontrol PID adalah :

$$PID = K_p \cdot e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt + K_p \cdot T_d \frac{de(t)}{dt} \quad (2.1)$$

Dimana:

PID = output dari pengontrol PID atau *Manipulated Variable*

$K_p$  = konstanta *Proporsional*

$K_i = K_p/T_i$

$K_d = K_p \cdot T_d$

$T_i$  = konstanta *Integral*

$T_d$  = konstanta *Derivatif*

$e(t)$  = error (selisih antara set point dengan level aktual)

Persamaan Pengontrol PID diatas dapat juga dituliskan sebagai berikut :

$$mv(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (2.2)$$

dengan:

$$K_i = K_p \times \frac{1}{T_i} \text{ dan } K_d = K_p \times T_d \quad (2.3)$$

Untuk lebih memaksimalkan kerja pengontrol diperlukan nilai batas minimum dan maksimum yang akan membatasi nilai *Manipulated Variable* yang dihasilkan.

Komponen kontrol PID ini terdiri dari tiga parameter yaitu *Proportional*, *Integratif* dan *Derivatif*. Ketiganya dapat dipakai bersamaan maupun sendiri-sendiri tergantung dari respon yang kita inginkan terhadap suatu plant. Untuk lebih

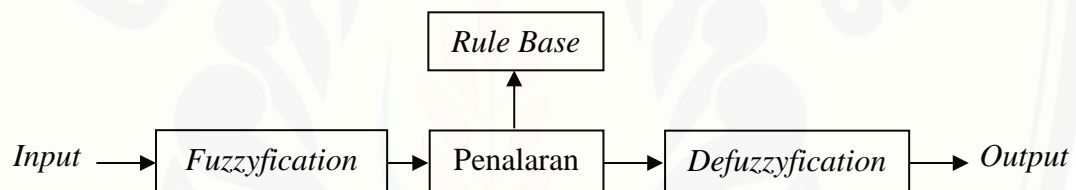
memaksimalkan kerja kontrol PID diperlukan pengaturan pada masing-masing parameter P, I dan D.

## 2.10 Kontrol Logika Fuzzy

### 2.10.1 Pengertian Logika Fuzzy

Logika *fuzzy* pertama kali dikenalkan pada tahun 1965 oleh Prof Lutfi A. Zadeh seorang peneliti pada bidang ilmu komputer di Universitas California di Barkley. Prof Lutfi A. Zadeh beranggapan logika benar dan salah tidak dapat mewakili setiap pemikiran manusia, kemudian dikembangkan ologika fuzzy yang dapat mempresentasikan setiap keadaan atau mewakili pemikiran manusia.

Logika *fuzzy* adalah logika samar. Dimana pada logika *fuzzy* suatu nilai dapat bernilai 'true' dan 'false' secara bersamaan. Logika *fuzzy* memiliki rentang antara 0 hingga 1, berbeda dengan logika digital yang hanya memiliki dua keanggotaan 0 atau 1 saja pada satu waktu.



Gambar 2.9 Blok Diagram Logika Fuzzy

Pada Gambar 2.9 menjelaskan proses dari logika *fuzzy*. Diawali dengan *input* kemudian dilakukan pembentukan *membership function* pada tahap *fuzzyfikasi*, setelah itu dilanjutkan dengan penalaran dimana dibentuk *rule base* untuk menentukan kondisi-kondisi yang dihasilkan oleh keluaran kontrol dan yang terakhir adalah tahap *defuzzyfikasi* yaitu mengkonversi setiap hasil dari *inference engine*.

### 2.10.2 Himpunan Fuzzy

Himpunan *fuzzy* merupakan pengelompokan berdasarkan variable *linguistik* yang dinyatakan dengan fungsi keanggotaan. Keanggotaan suatu himpunan nilainya dapat diketahui berdasarkan derajat keanggotaan yaitu antara 0 sampai dengan 1. (Fathor Rohman, 2017)

### 2.10.3 Fungsi Keanggotaan

Fungsi keanggotaan (*membership function*) adalah suatu kurva yang menunjukkan pemetaan titik input data ke dalam nilai keanggotaannya (sering juga disebut dengan derajat keanggotaan) yang memiliki interval antara 0 sampai 1.

Berikut nama-nama fungsi keanggotaan *Fuzzy Logic* :

1. Representasi Linear
2. Representasi Kurva Segitiga
3. Representasi Kurva Trapesium
4. Representasi Kurva Bentuk Bahu
5. Representasi Kurva-S
6. Representasi Bentuk Lonceng

### 2.10.4 Fuzzyfikasi

*Fuzzyfikasi* yaitu suatu proses untuk mengubah suatu masukan dari bentuk tegas (*crisp*) menjadi *fuzzy* (variabel linguistik) yang biasanya disajikan dalam bentuk himpunan fuzzy dengan suatu fungsi keanggotaannya masing-masing.

### 2.10.5 Rule Base

*Rule base* (aturan dasar) pada control logika *fuzzy* merupakan suatu bentuk aturan relasi “jika-maka” atau “*if-then*” seperti berikut ini: If x is A then y is B dimana A dan B adalah *linguistic values* yang didefinisikan dalam rentang variabel X dan Y. Pernyataan “y is B” disebut *consequent* atau kesimpulan.

### 2.10.6 Defuzzyfication

*Input* dari proses *defuzzyfikasi* adalah suatu himpunan *fuzzy* yang diperoleh dari komposisi aturan-aturan *fuzzy*, sedangkan *output* yang dihasilkan merupakan suatu bilangan pada domain himpunan *fuzzy* tersebut. Sehingga jika diberikan suatu himpunan *fuzzy* dalam *range* tertentu, maka harus dapat diambil suatu nilai *crisp* tertentu.

### BAB 3. METODOLOGI PENELITIAN

Pembahasan pada bab metode penelitian ini dijelaskan beberapa hal pokok yaitu obyek penelitian, waktu dan tempat kegiatan pada saat proses pembuatan dan pengambilan data, ruang lingkup, alat dan bahan yang digunakan, perancangan sistem, dan desain atau gambar dari alat yang digunakan.

#### 3.1 Waktu dan Tempat Penelitian

##### 3.1.1 Tempat

Adapun tempat penelitian, pengujian dan analisis dilakukan secara umum dilakukan di:

Tempat : Laboratorium Elektronika dan Terapan Fakultas Teknik Universitas Jember

Alamat : Jl. Slamet Riyadi No.62, Patrang, Kabupaten Jember.

##### 3.1.2 Waktu

Adapun waktu penelitian, pengujian dan analisis dilakukan secara pada Desember 2020 sampai dengan Mei 2020.

Tabel 3.1 *Timeline* Penelitian

No	Kegiatan	Januari				Februari				Maret				April				Mei				
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	
1	Studi Literatur																					
2	Perancangan Elektrikal dan Box Inkubator																					
3	Pembelian Alat dan Bahan																					
4	Pembuatan Inkubator Bayi																					
5	Pengujian																					
6	Analisis Data																					
7	Kesimpulan																					

### 3.2 Tahapan Penelitian

Langka-langkah yang akan digunakan dalam penelitian ini adalah sebagai berikut :

1. Studi Literatur.

Tahap pertama dengan mempelajari dan memahami teori-teori yang di peroleh dari beberapa buku, internet serta jurnal-jurnal dengan topik yang sama sehingga penulis dapat memahami konsep dan mendapatkan referensi dalam menyelesaikan masalah yang diteliti.

2. Melakukan Perancangan Perangkat Keras.

Tahap kedua merupakan perancangan perangkat keras dari alat dan bahan yang digunakan, membuat desain sistem inkubator, desain bentuk dan desain elektrik inkubator

3. Melakukan perancangan perangkat lunak.

Tahap ketiga adalah perancangan perangkat lunak yaitu perancangan kontrol *fuzzy*-PID, sensor, program dan Arduino pada Inkubator. Pada tahap ini dilakukan penentuan nilai  $K_p$ ,  $K_i$  dan  $K_d$  serta menentukan fungsi keanggotaan dari kontrol logika *fuzzy* yang akan digunakan pada inkubator agar mendapatkan hasil yang diinginkan.

4. Pengujian dan Pengambilan Data

Tahap keempat yaitu pengujian terhadap komponen-komponen seperti sensor DHT22, modul *driver* l298N, fan/kipas aksial, lampu dan modul ac *light dimmer*. Pengujian pada tahap ini sangatlah penting karena akan berpengaruh pada hasil.

5. Analisa Data dan Pembuatan Laporan

Tahap keenam adalah melakukan analisa data dari data yang telah didapatkan dari pengujian sebelumnya.

6. Kesimpulan

Tahap Ketujuh adalah tahap terakhir dari penelitian ini yaitu membuat kesimpulan dari pengujian dan analisa data yang diperoleh.

### 3.3 Perancangan Perangkat Keras

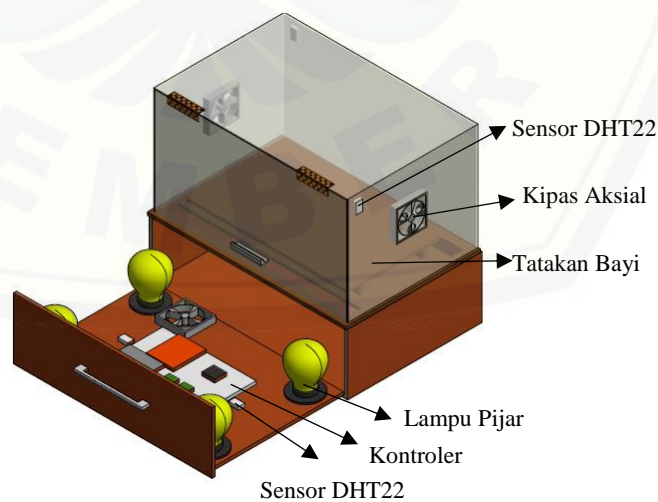
#### 3.3.1 Alat dan Bahan

Pada penelitian ini digunakan peralatan-peralatan sebagai penunjang pembuatan dan pengujian tugas akhir. Peralatan dan bahan yang digunakan dalam penelitian ini diantaranya adalah sebagai berikut:

- a. Sensor DHT22
- b. Inkubator Bayi
- c. Modul *AC Light Dimmer*
- d. Lampu Pijar
- e. Arduino UNO
- f. Laptop
- g. Fan/Kipas Aksial
- h. Modul *Driver L298N*
- i. LCD 16x2

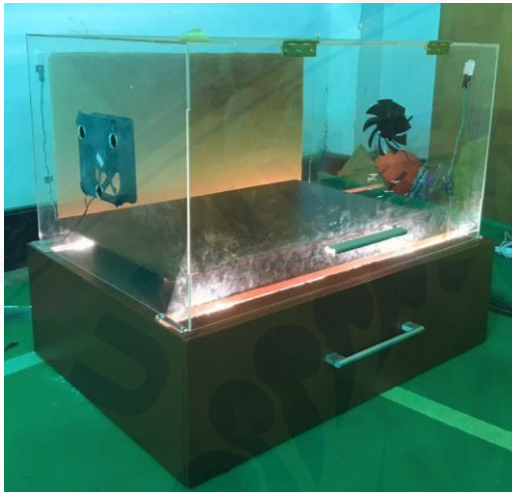
#### 3.3.2 Desain Alat

Inkubator dibagi menjadi dua bagian, yaitu *box* atas dan *box* bawah. Untuk bagian inkubator *box* bawah digunakan sebagai tempat rangkaian elektronika seperti pada Gambar 3.1 desain mekanik inkubator bayi prematur. Kemudian pada bagian inkubator *box* atas digunakan sebagai tempat menaruh bayi dan terdapat sensor DHT22 serta LCD 16x2.



Gambar 3.1 Desain Mekanik Inkubator Bayi

Inkubator menggunakan sensor DHT22 yang ditempatkan dibagian *box* atas dan *box* bawah pada inkubator. Sensor DHT22 ini berfungsi sebagai pendeteksi suhu dan kelembaban didalam inkubator. Kemudian digunakan fan sebagai pendingin apabila suhu melebihi batas yang ditentukan dengan modul *driver* L298N sebagai pengatur kecepatan putarannya dan digunakan modul *AC light dimmer* untuk menaikkan suhu dengan cara mengatur intensitas cahaya dari lampu.



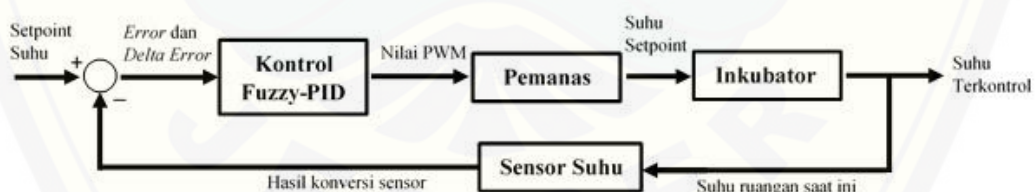
(a)



(b)

Gambar 3.2 (a) Inkubator Bayi, (b) Rangkaian Kontroler.

### 3.3.3 Rancang Bentuk dan Cara Kerja Alat



Gambar 3.3 Diagram Blok Inkubator Bayi

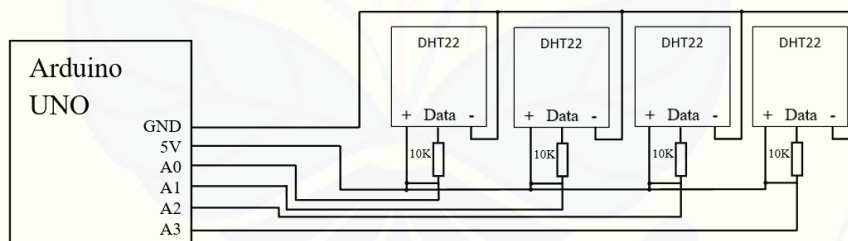
Pada Gambar 3.3 menjelaskan bahwa desain sistem menggunakan sistem pengendalian *close loop*. Sistem yang akan dirancang menggunakan kontrol *fuzzy-PID* sebagai sistem kendalinya. Nilai *setpoint* suhu ditentukan dari usia dan berat badan bayi. Kemudian digunakan *error* dan *delta error* sebagai parameter *membership function input* dan *output* suhu maupun kelembaban. Nilai *error* didapatkan dari selisih nilai antara *setpoint* dengan nilai *actual* atau *present value*

dan nilai *delta error* didapatkan dari selisih nilai antara *error* pada waktu  $t$  dengan *error* pada waktu  $t-1$ . *Present value* adalah nilai suhu yang didapatkan dari hasil pembacaan sensor DHT22. Setelah melalui pengolahan kontrol oleh Fuzzy-PID pada *mikrokontroler* Arduino uno maka akan dihasilkan pengambilan keputusan berupa *output* nilai PWM untuk mengatur kecerahan lampu. Setelah itu akan terjadi proses pada inkubator berupa perubahan suhu sesuai dengan *setpoint* yang telah ditentukan. Setelah suhu sesuai maka mikrokontroler akan mematikan lampu apabila terjadi perubahan suhu pada inkubator maka sistem akan beroperasi kembali untuk mencapai *setpoint* yang telah diinput-kan.

### 3.3.4 Desain Elektrik

#### 3.3.4.1 Rangkaian Sensor DHT22

Sensor DHT22 dihubungkan dengan mikrokontroler Arduino Uno. Komunikasi antara sensor DHT22 dengan Arduino Uno menggunakan pin A0, A1, A2 dan A3 sebagai input karena digunakan 4 (empat) buah sensor. Kemudian untuk catu daya dihubungkan dengan pin *power* 5V dan *ground*. Gambar 3.4 menunjukkan tampilan interkoneksi antara system Arduino dengan sensor DHT22.



Gambar 3.4 Rangkaian DHT22 dengan Arduino

```

1 | #include <DHT.h>
2 | DHT dht(A0, DHT22); //Pin, Jenis DHT
3 | DHT dht2(A1, DHT22);
4 | DHT dht3(A2, DHT22);
5 | DHT dht4(A3, DHT22);
6 | float kelembaban1 = dht.readHumidity();
7 | float kelembaban2 = dht2.readHumidity();

```

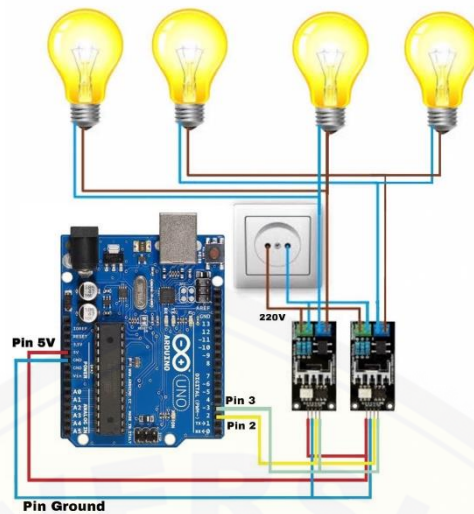


```
8 | float kelembaban3 = dht3.readHumidity();
9 | float kelembaban4 = dht4.readHumidity();
10| float suhu1 = dht.readTemperature();
11| float suhu2 = dht2.readTemperature();
12| float suhu3 = dht3.readTemperature();
13| float suhu4 = dht4.readTemperature();
```

Pada baris pertama dalam *listing* program arduino merupakan *library* dari sensor DHT. Pada baris ke-2 sampai baris ke-5 merupakan inisialisasi dari variabel pin dan jenis DHT yang digunakan, dalam penelitian ini digunakan 4 (empat) sensor DHT22 sehingga terdapat 4 (empat) pin yang digunakan sebagai *input*. Pada baris ke-6 sampai baris ke-9 digunakan untuk menyimpan nilai kelembaban yang terbaca oleh sensor kedalam variabel *kelembaban1* sampai *kelembaban4* dengan tipe data desimal (*float*). Kemudian pada baris ke-10 sampai baris ke-13 digunakan untuk menyimpan nilai suhu yang terbaca oleh sensor kedalam variabel *suhu1* sampai *suhu4* dengan tipe data desimal (*float*).

#### 3.3.4.2 Rangkaian *Heater* dengan Modul AC *Light Dimmer*

*Heater* yang merupakan sebuah lampu pijar pada perancangan ini langsung terhubung dengan pin *load* pada modul AC *light dimmer*. Sedangkan modul AC *light dimmer* dihubungkan dengan Arduino Uno dan sumber tegangan sebesar 220V. Komunikasi antara modul AC *light dimmer* dengan Arduino uno menggunakan pin 2 dan pin 9 sebagai *output* PWM. Selain dihubungkan dengan sumber tegangan sebesar 220V, modul ini juga dihubungkan dengan pin *power* 5V dan *ground* pada Arduino uno. Gambar 3.5 menunjukkan rangkaian *heater* dengan modul AC *light dimmer* dan Arduino uno, dimana dari setiap modul ac *light dimmer* yang digunakan terdapat 2 lampu yang diseri.



Gambar 3.5 Rangkaian *Heater* dengan Modul *AC Light Dimmer* dan Arduino Uno.

```

1| #include "RBDdimmer.h"
2| #define USE_SERIAL Serial
3| #define outputPin 3
4| #define zerocross 2

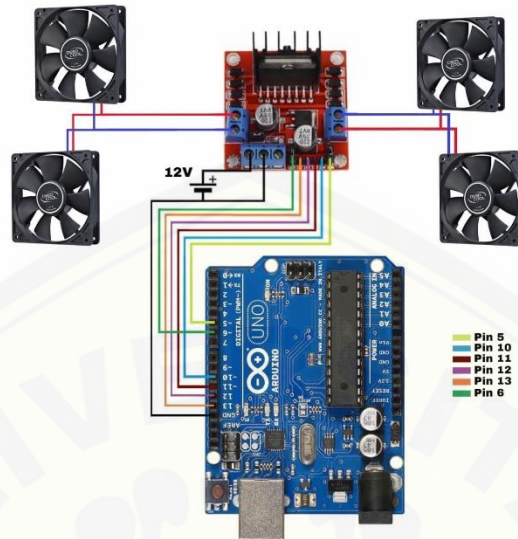
```

Pada baris ke-1 *listing* program Arduino merupakan *library* dari modul *AC light dimmer*, pada baris ke-2 merupakan inisialisasi dari variabel *USE\_SERIAL* yang digunakan dari modul *AC light dimmer*, pada baris ke-3 merupakan inisialisasi dari variabel *outputPin* dari modul *AC light dimmer* dan pada baris ke-4 merupakan inisialisasi dari variabel *zerocross* dari modul *AC light dimmer*.

#### 3.4.4.3 Rangkaian Fan/Kipas Aksial dengan Modul *Driver* L289N

Fan/Kipas Aksial langsung terhubung dengan pin *output* A dan B pada modul *driver* L298N yang berfungsi sebagai pengontrol menggunakan nilai PWM dengan cara melepas *jumper* pada pin *enable* A dan B. Kemudian modul *driver* L298N dihubungkan dengan Arduino uno pada pin 10, 11, 12 dan 13 sebagai *input* serta pin 5 dan 6 sebagai *output*. Selain itu untuk catu daya dihubungkan dengan pin *power* 5V dan *ground*. Gambar 3.6 menunjukkan rangkaian kipas aksial dengan

modul *driver* L298n dan Arduino uno, digunakan 2 kipas aksial disetiap pin *output* yang ada pada modul.

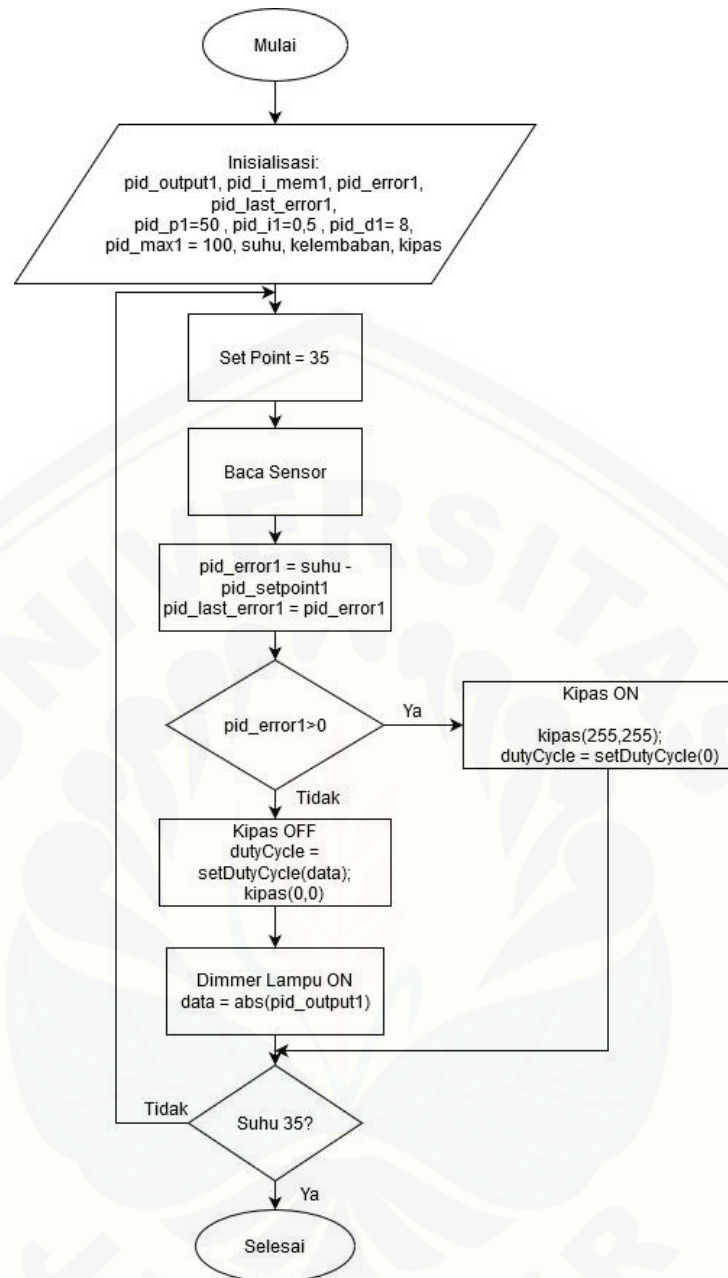


Gambar 3.6 Rangkaian Kipas Aksial dengan Modul *Driver* L298N dan Arduino Uno.

### 3.5 Perancangan Perangkat Lunak

#### 3.5.1 Sistem Kontrol PID

Langkah pertama yang dilakukan dalam sistem kontrol PID adalah perancangan kontrol PID, dalam perancangannya dibutuhkan algoritma yang sesuai agar dapat mendapatkan hasil control yang sesuai. Nantinya algoritma dari perancangan kontrol PID akan dimasukkan kedalam Arduino. Pada kontrol PID terdapat parameter  $K_p$ ,  $K_i$  dan  $K_d$  sebagai penentu keluaran atau hasil dari respon sistem, selain itu menentukan nilai  $K_p$ ,  $K_i$  dan  $K_d$  adalah hal yang paling penting dari kontrol PID. Pengujian akan menggunakan metode *trial and error*.



Gambar 3.7 Flowchart Program Kontrol PID

Pada Gambar 3.7 merupakan *flowchart* dari program kontrol PID, dimana `pid_output1` merupakan keluaran dari kontrol PID untuk mengatur pemanas. Sedangkan `kipas(255,255)` untuk mengatur kecepatan dari kipas aksial ketika diketahui suhu inkubator melebihi suhu pada *set point*.

Dalam penelitian ini dilakukan pengujian menggunakan metode manual *tuning*. Manual *tuning* merupakan metode untuk mencari parameter-parameter

kontrol PID dengan cara *trial and error*. Langkah-langkah yang dilakukan dalam mencari parameter-parameter dengan manual tuning adalah sebagai berikut.

1. Memberikan nilai pada parameter  $K_p$  untuk mengontrol keluaran *plan* dalam mencapai *setpoint*.
2. Memberikan nilai pada parameter  $K_i$  untuk mengontrol keluaran *plan* dalam mencapai *setpoint*. Namun pemberian parameter ini akan meningkatkan *overshoot* dan *settling time* meningkat.
3. Memberikan nilai pada parameter  $K_d$  untuk mengontrol keluaran *plan* dalam mencapai *setpoint*.

### 3.5.2 Sistem Logika Fuzzy

Pada proses pengambilan keputusan dengan kontrol logika *fuzzy* yang digunakan dalam sistem inkubator bayi dilakukan melalui beberapa tahapan, yaitu tahap *fuzzyfikasi* yang merupakan proses pembentukan *membership function*, kemudian tahap *inferensi* yang merupakan proses pembentukan *rule base* dan yang terakhir tahap *defuzzyfikasi*.

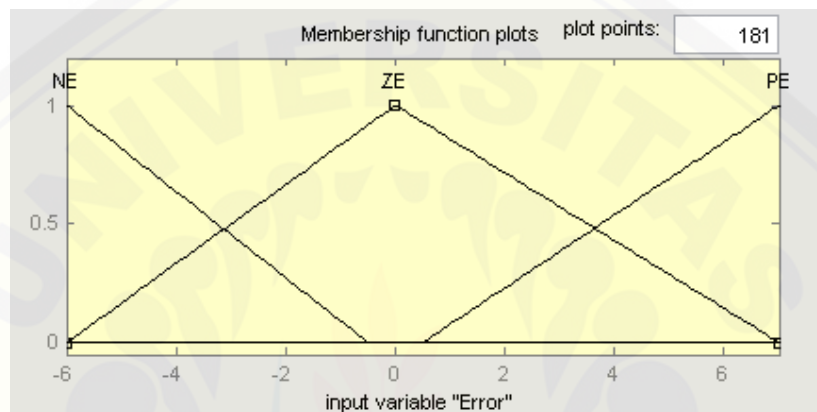
Algoritma logika *fuzzy* memiliki beberapa metode dalam merepresentasikan hasil logikanya, yaitu metode Tsukamoto, Sugeno dan Mamdani. Metode Mamdani yang dipilih karena lebih spesifik dalam prosesnya, metode ini lebih memperhatikan kondisi yang akan terjadi untuk setiap daerah *fuzzynya* sehingga dapat menghasilkan keputusan yang lebih akurat.

#### 1. Tahapan Fuzzyfikasi

Tahapan *fuzzyfikasi* ini dilakukan pembentukan *membership function*. *Membership function* ini sangat berperan penting dalam sistem. Karena elemen keanggotaan ini berfungsi sebagai *input* dan *output* untuk nilai yang dihasilkan inkubator bayi prematur. Parameter masukan berupa *error* dan *delta error*.

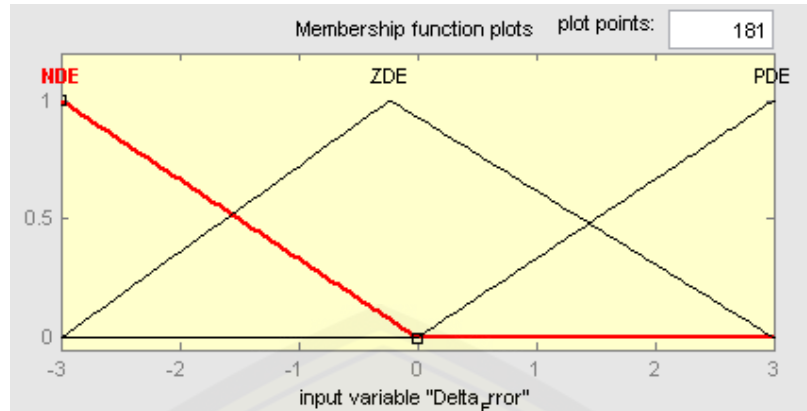
Pada pembentukan *membership function* ini terlebih dahulu ditentukan besarnya *setpoint* untuk membuat *membership function* dan pada proses perhitungan *error* dan *delta error*. Untuk besarnya nilai *setpoint* yang digunakan yaitu 32°C, 33°C, 34°C dan 35°C dimana nilai ini merupakan suhu yang dibutuhkan oleh bayi prematur berdasarkan dari usia dan berat badannya.

Pada perancangan sistem kontrol ini bertujuan untuk mengontrol suhu. Kontrol pada sistem ini berdasarkan dari *error* suhu dan *delta error* suhu. Nantinya sensor akan mengukur suhu dan kelembaban sebenarnya pada inkubator dan mengirimnya ke mikrokontroler, kemudian mikrokontroler akan menghitung selisih suhu pada *setpoint* dengan suhu sebenarnya pada inkubator yang kemudian menghasilkan *error* ( $e$ ) dan mencatatnya. Kemudian mikrokontroler akan menghitung selisih *error* ( $e$ ) pada waktu  $t$  dengan nilai ( $e$ ) pada waktu  $t-1$  dan menghasilkan *delta error* ( $de$ ) atau perubahan *error*.



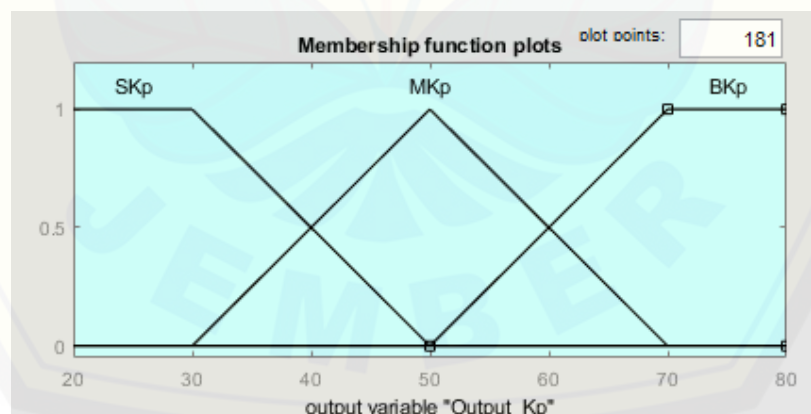
Gambar 3.8 *Membership Function Input Error Suhu*

Pada Gambar 3.8 penentuan *input error* digunakan beberapa kemungkinan yang terjadi untuk besarnya nilai *error* yang nantinya akan dijadikan sebagai batas terbesar dan juga batas terkecil untuk daerah batas *crisp*. Dari beberapa nilai *error* yang terjadi terdapat kemungkinan nilai *error* tertinggi yaitu 7 yaitu ketika sensor suhu masih membaca pada suhu ruang yakni 28°C. Sedangkan untuk kemungkinan nilai *error* terkecil adalah -6 ketika pembacaan sensor sebesar 38°C dengan *setpoint* 32°C. Untuk pemberian *label* pada *membership function* dalam *input error* yaitu menggunakan NE (*Negatif Error*), ZE (*Zero Error*) dan PE (*Positif Error*).



Gambar 3.9 *Membership Function Input Delta Error Suhu*

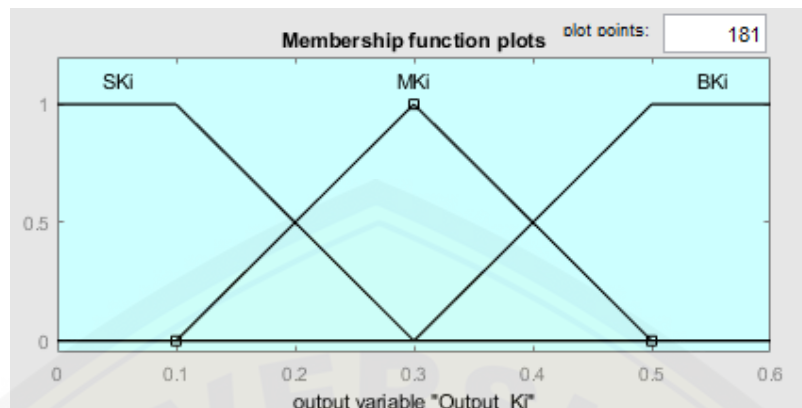
Kemudian pada Gambar 3.9 penentuan *input delta error* digunakan beberapa kemungkinan yang terjadi untuk besarnya nilai *delta error* yang nantinya akan dijadikan sebagai batas terbesar dan juga batas terkecil untuk daerah batas *crisp*. Dari beberapa nilai *delta error* yang terjadi terdapat kemungkinan nilai *delta error* tertinggi yaitu 3, sedangkan untuk kemungkinan nilai *delta error* terkecil adalah -3. Untuk pemberian *label* pada *membership function* dalam *input error* yaitu menggunakan NDE (*Negatif Delta Error*), ZE (*Zero Delta Error*) dan PDE (*Positif Delta Error*).



Gambar 3.10 *Membership Function Output Kp*

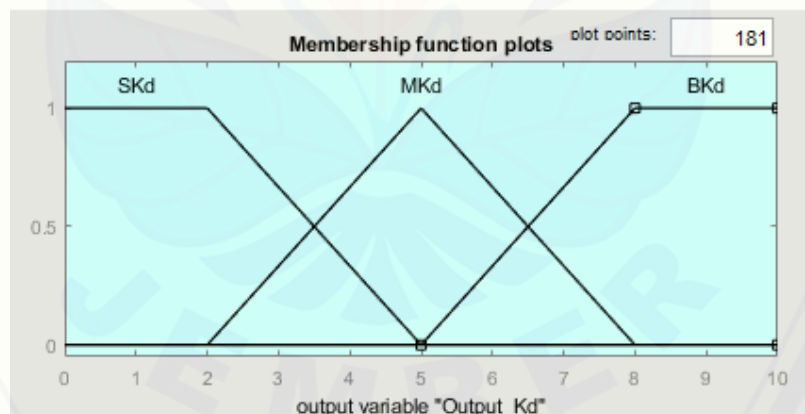
Dalam Gambar 3.10 penentuan *membership function* untuk *output Kp* menggunakan nilai *Kp* dari pengujian pada kontrol P. Untuk penamaan *label* pada *membership function* dari *output Kp* yaitu menggunakan SKp (*Small Kp*), MKp

(*Medium Kp*) dan BKp (*Big Kp*). Pada *output Kp* daerah batas *crisp* yang digunakan adalah 20 sampai dengan 80.



Gambar 3.11 *Membership Function Output Ki*

Dalam Gambar 3.11 penentuan *membership function* untuk *output Ki* menggunakan nilai  $K_i$  dari pengujian pada kontrol PI. Untuk penamaan *label* pada *membership function* dari *output Ki* yaitu menggunakan SKi (*Small Ki*), MKi (*Medium Ki*) dan BKi (*Big Ki*). Pada *output Ki* daerah batas *crisp* yang digunakan adalah 0 sampai dengan 0,6.



Gambar 3.12 *Membership Function Output Kd*

Dalam Gambar 3.12 penentuan *membership function* untuk *output Kd* menggunakan nilai  $K_d$  dari pengujian pada kontrol PD. Untuk penamaan *label* pada *membership function* dari *output Kd* yaitu menggunakan SKd (*Small Kd*), MKd (*Medium Kd*) dan BKd (*Big Kd*). Pada *output Kd* daerah batas *crisp* yang digunakan adalah 0 sampai dengan 10.



## 2. Tahapan *Inference*

Pada tahapan ini dilakukan pembuatan aturan dari setiap kondisi yang terjadi dalam suatu proses, dengan cara memetakan nilai derajat keanggotaan dari masing-masing *error* dan *delta error* kedalam sebuah tabel yang berisi basis aturan yang dipilih atau biasa disebut *rule base* dan digunakan sebuah metode untuk pemilihan nilai dari derajat keanggotaan.

Tabel 3.2 *Rule Output Kp Suhu*

Delta Error \ Error	NE	ZE	PE
NDE	SKp	MKp	BKp
ZDE	SKp	SKp	BKp
PDE	SKp	MKp	BKp

Tabel 3.2 merupakan pengelompokan kondisi-kondisi untuk *output* dari Kp. Terdapat 9 kondisi *input* dan *output* untuk Kp. Digunakan tiga kondisi untuk *Error* dan *Delta Error*.

Tabel 3.3 *Rule Output Ki Suhu*

Delta Error \ Error	NE	ZE	PE
NDE	Ski	MKi	BKi
ZDE	Ski	MKi	BKi
PDE	Ski	MKi	BKi

Tabel 3.3 merupakan pengelompokan kondisi-kondisi untuk *output* dari Ki. Terdapat 9 kondisi *input* dan *output* untuk Ki. Digunakan tiga kondisi untuk *Error* dan *Delta Error*.

Tabel 3.3 *Rule Output Kd Suhu*

Delta Error \ Error	NE	ZE	PE
NDE	SKd	MKd	BKp
ZDE	SKd	SKd	MKp
PDE	SKd	MKd	BKp

Tabel 3.3 merupakan pengelompokan kondisi-kondisi untuk *output* dari Kd. Terdapat 9 kondisi *input* dan *output* untuk Kd. Digunakan tiga kondisi untuk *Error* dan *Delta Error*.

Keterangan:

NE = Negatif Error

NDE = Negatif Delta Error

ZE = Zero Error

ZDE = Zero Delta Error

PE = Positif Error

PDE = Positif Delta Error

### 3. Tahapan *Defuzzifikasi*

Pada tahap ini merupakan tahapan terakhir dalam suatu sistem logika *fuzzy* dimana tujuannya adalah mengkonversi setiap hasil dari *inference* dalam bentuk *rule base* kedalam suatu bilangan *real* dan hasil dari konversi tersebut adalah sebuah keputusan kontrol.

```
void Defuzzification_Kp(){
SKp=max(Fuzzy_Kp[0][0],max(Fuzzy_Kp[1][0],
max(Fuzzy_Kp[2][0], Fuzzy_Kp[1][1])));
MKp=max(Fuzzy_Kp[0][1],Fuzzy_Kp[2][1]);
BKp=max(Fuzzy_Kp[0][2],max(Fuzzy_Kp[1][2],Fuzzy_Kp[2][2]
));
if(!(SKp==0 && MKp==0 && BKp==0)){
hasil_Kp=(SKp*OSKp+MKp*OMKp+BKp*OBKp)/(SKp+MKp+BKp);
}
}
```

```
void Defuzzification_Ki(){
SKi=max(Fuzzy_Ki[0][2],max(Fuzzy_Ki[1][2],Fuzzy_Ki[2][2]
));
MKi=max(Fuzzy_Ki[0][1],max(Fuzzy_Ki[1][1],Fuzzy_Ki[2][1]
));
```

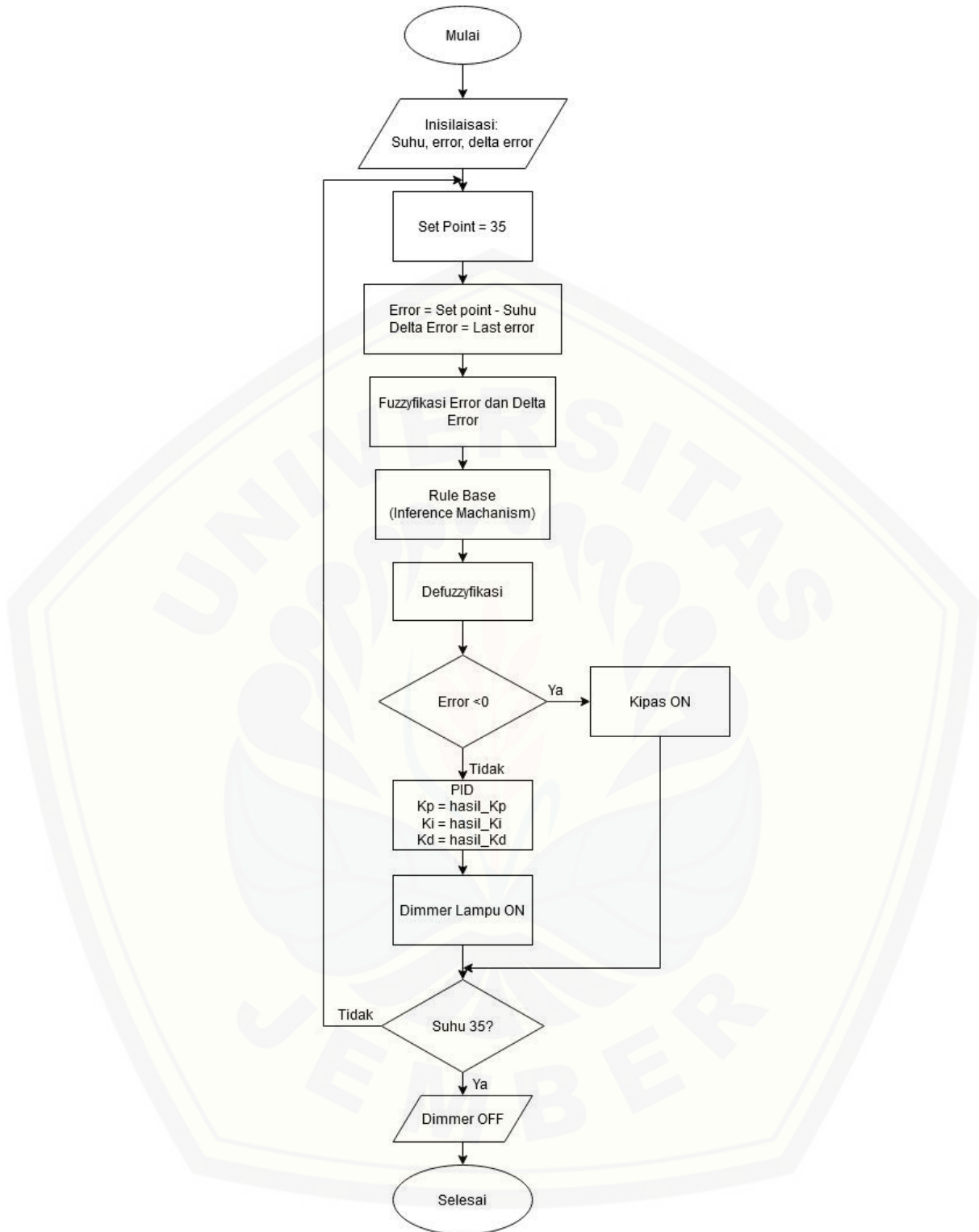
```

BKi=max(Fuzzy_Ki[0][0],max(Fuzzy_Ki[1][0],Fuzzy_Ki[2][0
])));
    if(!(SKi==0 && MKi==0 && BKi==0)){
        hasil_Ki=(SKi*OSKi+MKi*OMKi+BKi*OBKi)/(SKi+MKi+BKi);
    }
}

void Defuzzification_Kd(){
SKd=max(Fuzzy_Kd[0][0],max(Fuzzy_Kd[1][0],max(Fuzzy_Kd[
2][1],Fuzzy_Kd[1][1])));
MKd=max(Fuzzy_Kd[0][1],max(Fuzzy_Kd[2][1],Fuzzy_Kd[1][2
]));
BKd=max(Fuzzy_Kd[0][2],Fuzzy_Kd[2][2]);
    if(!(SKd==0 && MKd==0 && BKd==0)){
        hasil_Kd=(SKd*OSKd+MKd*OMKd+BKd*OBKd)/(SKd+MKd+BKd);
    }
}

```

Pada tahap ini dilakukan pengelompokan kondisi dalam pengambilan keputusan kontrol berdasarkan dari *rule base* yang telah dibentuk pada tahapan inferensi. Pada *void* pertama dalam program dilakukan pengelompokan kondisi untuk *output* K<sub>p</sub>, pada *void* kedua dilakukan pengelompokan kondisi untuk *output* K<sub>i</sub> dan pada *void* ketiga dilakukan pengelompokan kondisi untuk *output* K<sub>d</sub>.



Gambar 3.13 Flowchart Kontrol Fuzzy-PID

Pada Gambar 3.13 menunjukkan proses dari kontrol fuzzy-PID diawali dengan inisialisasi suhu, kemudian masukan *setpoint* dan perhitungan *error* dan

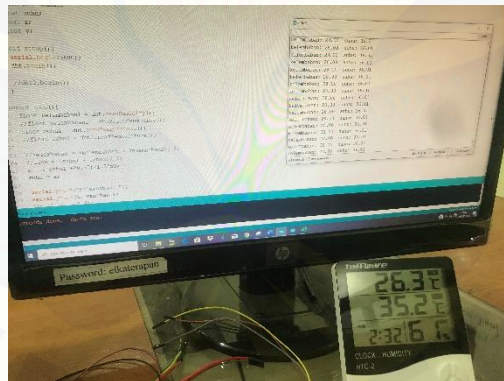
*delta error*. Setelah itu dilakukan proses *fuzzyfikasi*, inferensi dan *defuzzyfikasi*. Dilanjutkan dengan pengambilan keputusan untuk nilai *error* yang didapatkan, jika nilai *error* kurang dari nol maka kipas akan aktif sedangkan jika nilai *error* lebih dari nol maka akan dilakukan proses *hasil\_kp*, *hasil\_ki* dan *hasil\_kd* untuk menyalakan pemanas.

### 3.6 Metode Pengujian dan Pengumpulan Data

Pengujian dilakukan dengan cara menguji sensor DHT22, modul AC *light dimmer* dengan lampu pijar, modul *driver* l298n dengan kipas aksial. Pengujian pemanas pada inkubator bayi menggunakan sistem kontrol *fuzzy*-PID untuk mengatur suhu.

#### 3.6.1 Pengujian Sensor DHT22

Tahap pertama dalam pengujian sensor DHT22 dilakukan pengambilan data sensor dengan cara membandingkan hasil pembacaan sensor DHT22 dengan hasil pembacaan termometer digital. Proses pengambilan data pada pengujian sensor DHT22 dilakukan seperti pada Gambar 3.14.



Gambar 3.14 Pengujian Sensor

#### 3.6.2 Pengujian Kontrol PID

Pada pengujian kontrol PID dilakukan tuning dengan menggunakan *trial and error*. Pengujian pertama dilakukan pengujian untuk kontrol P seperti pada Gambar 3.15, dimana nilai kontrol P diubah-ubah sedangkan nilai untuk kontrol I dan kontrol D adalah nol. Setelah mendapatkan hasil terbaik untuk kontrol P, dilanjutkan dengan pengujian kontrol PI seperti pada Gambar 3.16, dimana nilai

kontrol P adalah nilai dengan hasil respon terbaik pada pengujian sebelumnya sedangkan nilai untuk kontrol I diubah-ubah dan nilai untuk kontrol D adalah nol. Kemudian dilanjutkan dengan pengujian kontrol PD seperti pada Gambar 3.17, dimana nilai kontrol P adalah nilai dengan hasil respon terbaik pada pengujian sebelumnya sedangkan untuk kontrol D diubah-ubah dan nilai untuk kontrol I adalah nol. Untuk pengujian selanjutnya adalah pengujian kontrol PID seperti pada Gambar 3.18, dimana nilai-nilai yang digunakan untuk kontrol P, I dan D merupakan nilai-nilai yang telah diujikan pada pengujian sebelumnya.

```

// PID NEW_21_ajp1 (Arduino 1.8.12)
// File Edit Sketch Tools Help

// PID NEW_21_ajp1
#include <PID.h>
#define PID_OUT_PIN 4
#define PID_IN_PIN 3
#define PID_SETPOINT 100
#define PID_OUTPUT_PIN 5
#include <LiquidCrystal_I2C.h>

// Set the LCD address to 0x27 for a 16 chars and 2 line display
LiquidCrystal_I2C lcd(0x27, 16, 2);

// PID Controller
PID myPID(0, 1.0, 0.0, 0.0);

// PID Controller
// Specify the PID and output pin numbers
void setup() {
  pinMode(PID_OUT_PIN, OUTPUT);
  pinMode(PID_IN_PIN, INPUT);
  pinMode(PID_OUTPUT_PIN, OUTPUT);
  lcd.begin(16, 2);
  myPID.begin();
}

void loop() {
  float out = myPID.Compute();
  digitalWrite(PID_OUT_PIN, out);
  digitalWrite(PID_OUTPUT_PIN, out);
  lcd.setCursor(0, 0);
  lcd.print("PID NEW_21_ajp1");
  lcd.setCursor(0, 1);
  lcd.print("Setpoint: 100");
  lcd.print("Output: ");
  lcd.print(out);
  delay(100);
}
  
```

Gambar 3.15 Pengujian Kontrol P

```

// PID NEW_21_ajp1 (Arduino 1.8.12)
// File Edit Sketch Tools Help

// PID NEW_21_ajp1
#include <PID.h>
#define PID_OUT_PIN 4
#define PID_IN_PIN 3
#define PID_SETPOINT 100
#define PID_OUTPUT_PIN 5
#include <LiquidCrystal_I2C.h>

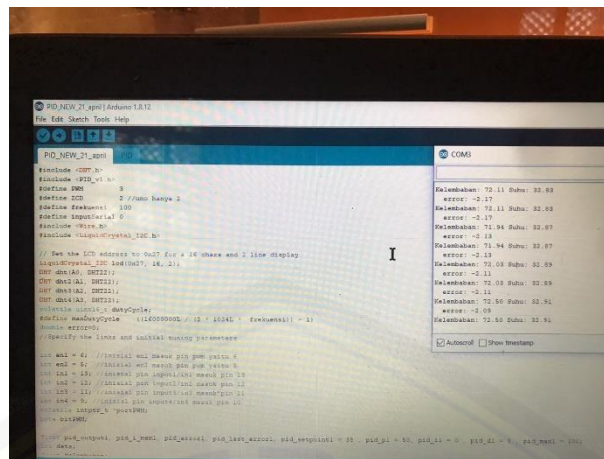
// Set the LCD address to 0x27 for a 16 chars and 2 line display
LiquidCrystal_I2C lcd(0x27, 16, 2);

// PID Controller
PID myPID(0, 1.0, 0.0, 0.0);

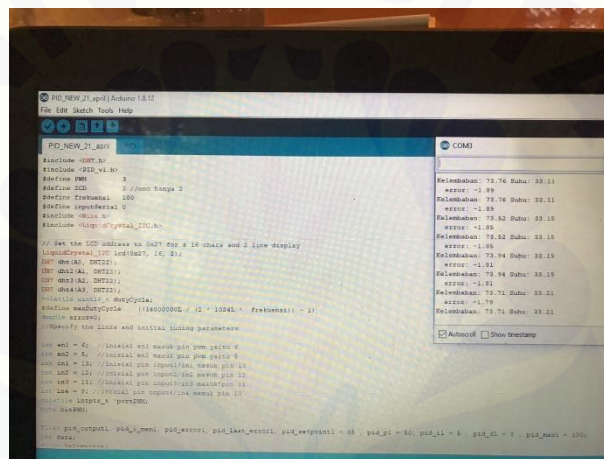
// PID Controller
// Specify the PID and output pin numbers
void setup() {
  pinMode(PID_OUT_PIN, OUTPUT);
  pinMode(PID_IN_PIN, INPUT);
  pinMode(PID_OUTPUT_PIN, OUTPUT);
  lcd.begin(16, 2);
  myPID.begin();
}

void loop() {
  float out = myPID.Compute();
  digitalWrite(PID_OUT_PIN, out);
  digitalWrite(PID_OUTPUT_PIN, out);
  lcd.setCursor(0, 0);
  lcd.print("PID NEW_21_ajp1");
  lcd.setCursor(0, 1);
  lcd.print("Setpoint: 100");
  lcd.print("Output: ");
  lcd.print(out);
  delay(100);
}
  
```

Gambar 3.16 Pengujian Kontrol PI



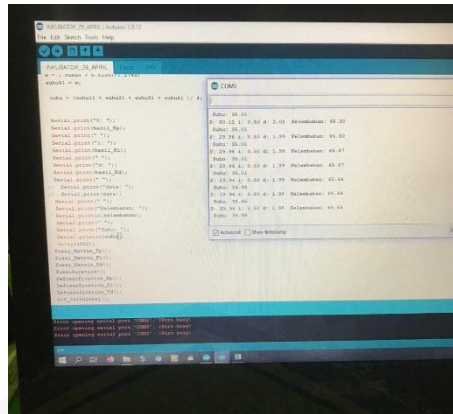
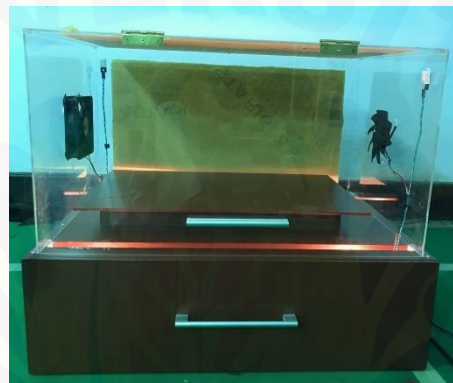
Gambar 3.17 Pengujian Kontrol PD



Gambar 3.18 Pengujian Kontrol PID

### 3.6.3 Pengujian Kontrol *Fuzzy*-PID

Untuk pengujian kontrol *fuzzy*-PID merupakan pengujian keseluruhan dalam penelitian ini, dalam pengujian ini digunakan dua analisis yakni pengujian tanpa beban seperti pada Gambar 3.20 dan pengujian dengan beban seperti pada Gambar 3.21, pada pengujian dengan beban digunakan air dengan suhu 36,5°C-37,5°C dimana suhu tersebut menyerupai suhu dari bayi yang baru lahir.

Gambar 3.19 Pengujian Kontrol *Fuzzy*-PIDGambar 3.20 Pengujian Tanpa Beban dengan Kontrol *Fuzzy*-PIDGambar 3.21 Pengujian dengan Beban dengan Kontrol *Fuzzy*-PID

### 3.7 Metode dan Analisa

Analisis dari inkubator bari otomatis menggunakan metode kontrol *fuzzy*-PID diharapkan memiliki kelebihan respon sistem yang cepat dalam menghangatkan inkubator bayi, selain itu juga diharapkan memiliki tingkat



keamanan yang baik dengan nilai *overshoot* yang rendah dan pemanas yang tidak membahayakan bayi.

Pada pengujian inkubator bayi digunakan media air dengan suhu yang menyerupai bayi baru lahir, yakni  $36,5^{\circ}\text{C}$ - $37,5^{\circ}\text{C}$ . Pengujian dengan menambahkan media air sebagai beban guna mengetahui respon sistem dari kontrol *fuzzy*-PID untuk mencapai *setpoint* ketika dipengaruhi suhu lain selain suhu ruangan didalam inkubator dan suhu ruangan diluar inkubator. Dalam melakukan pengujian digunakan waktu selama 7 menit disetiap *setpoint* yang digunakan, hal ini dilakukan untuk mempermudah dalam membandingkan hasil dari keluaran kontrol.



## BAB 5. PENUTUP

### 5.1 Kesimpulan

Dari hasil yang diperoleh pada pengujian-pengujian dalam penelitian yang telah dilakukan, dapat diambil beberapa kesimpulan sebagai berikut:

1. Sistem kontrol fuzzy-PID memiliki keluaran yang lebih baik dan lebih stabil dibandingkan dengan kontrol PID dalam mengatur suhu, dibuktikan pada hasil respon kontrol fuzzy-PID ketika setpoint 35°C memiliki rise time yang lebih cepat dibandingkan dengan hasil respon kontrol PID yakni 205 detik, selain itu hasil respon kontrol fuzzy-PID pada setpoint 35°C juga memiliki settling time yang lebih baik jika dibandingkan dengan hasil respon kontrol PID yakni 227 detik.
2. Penggunaan lampu pijar sebagai pemanas menghasilkan *overshoot* yang rendah sehingga aman untuk bayi lahir prematur, selain itu dengan menggunakan lampu pijar dapat mempermudah konsumen dalam menggantinya serta harganya terjangkau dan mudah ditemukan. Pembuatan dari inkubator bayi premature ini kurang lebih sama dengan harga inkubator local yang ada dipasaran yakni ± Rp 1.620.000.

### 5.2 Saran

Dengan harapan penelitian ini memiliki hasil yang lebih baik dan lebih maksimal dikemudian hari, penulis memberikan saran sebagai berikut:

1. Pada inkubator bayi ini belum terdapat tombol untuk mengatur set point guna mempermudah pengguna inkubator bayi.
2. Pada inkubator bayi ini belum terdapat pengaturan kelembaban secara otomatis.

DAFTAR PUSTAKA

- Budiono. 2013. “Desain dan Pembuatan Inkubator Berbasis Mikrokontroler Dengan Logika Fuzzy”. Jurnal Gamma. Jurusan Teknik Mesin.
- Komang Suchy, dkk. 2016. “Rancang Bangun Sistem Kontrol Suhu dan Kelembaban pada Inkubator Bayi Dengan Modul Thermoelectric dan Metode Fuzzy Logic”. Jurnal, Prodi S1 Teknik Elektro.
- Huang, Y., & Yasunobu, S. (2000, May). A general practical design method for fuzzy PID control from conventional PID control. In *Ninth IEEE International Conference on Fuzzy Systems. FUZZ-IEEE 2000 (Cat. No. 00CH37063)* (Vol. 2, pp. 969-972). IEEE.
- Noor Yulita, dkk. 2016. “Control Temperature on Plant Baby Incubator With Fuzzy Logic”. Jurnal SIMETRIS. Fakultas Teknik. Diakses pada 30 Oktober 2019.
- Wahyumithali Hameliani. 2016. “Kontrol Suhu dan dan Kelembaban Pada Mesin Pengembang Roti Menggunakan Metode Fuzzy-PID”. Jurnal. Program Studi Teknik Mekatronika.
- Freer, Y., & Lyon, A. (2012). Temperature monitoring and control in the newborn baby. *Paediatrics and Child Health*, 22(4), 127-130.
- Pratiwi, Dimas Putri. 2014. “Pemantau dan Pengatur Suhu Inkubator Berbasis Wifi”. Skripsi. Diterbitkan. Universitas Telkom: Bandung.
- Budiyanta, N. E., Tanudjaja, H., & Mulyadi, M. (2019). Rancang Bangun Robot Line Follower Portable Sebagai Upaya Minimalisasi Sampah Elektronik di Ranah Robotika. *TESLA: Jurnal Teknik Elektro*, 20(2), 148-156.
- Duan, X. G., Deng, H., & Li, H. X. (2012). A saturation-based tuning method for fuzzy PID controller. *IEEE Transactions on Industrial Electronics*, 60(11), 5177-5185.

- Priyandono, B. (2013). Analisis Konservasi Energi Listrik pada Rumah Tinggal Daya 2200VA dengan Beban Penerangan. *Jurnal Online Sekolah Tinggi Teknologi Mandala*, 6(1), 23-32.
- Mendel, J. M. (2007). Type-2 fuzzy sets and systems: an overview. *IEEE computational intelligence magazine*, 2(1), 20-29.
- Rohman, F. 2017. Rancang Bangun Alat Pengering Tembakau Otomatis Dengan Metode Pengolahan Citra Digital Menggunakan Kontrol Hybrid Fuzzy-Pid.
- Setyaningsih, N. Y. D., & Murti, A. C. (2016). Control Temperature On Plant Baby Incubator With Fuzzy Logic. *Simetris: Jurnal Teknik Mesin, Elektro dan Ilmu Komputer*, 7(1), 273-282.
- Arulmozhiyal, R., & Kandiban, R. (2012, January). Design of fuzzy PID controller for brushless DC motor. In *2012 International Conference on Computer Communication and Informatics* (pp. 1-7). IEEE.

## LAMPIRAN

**1. Listing Program Arduino Uno**

```

#include <DHT.h>
#define PWM          3
#define ZCD          2
#define frekuensi    100
#define inputSerial  0
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

// Set the LCD address to 0x27 for a 16 chars and 2 line
display
LiquidCrystal_I2C lcd(0x27, 16, 2);
DHT dht(A0, DHT22);
DHT dht2(A1, DHT22);
DHT dht3(A2, DHT22);
DHT dht4(A3, DHT22);
volatile uint16_t dutyCycle;
#define maxDutyCycle ((16000000L / (2 * 1024L *
frekuensi)) - 1)

float NE, ZE, PE;
float NDE, ZDE, PDE;
float SKp, MKp, BKp;
float SKi, MKi, BKi;
float SKd, MKd, BKd;
float Fuzzy_Kp[3][3];
float Fuzzy_Ki[3][3];
float Fuzzy_Kd[3][3];
float OSKp=30, OMKp=50, OBKp=70;
float OSKi=0.1, OMKi=0.3 , OBKi=0.5;
float OSKd=2, OMKd=5 , OBKd=8;

```

```
float hasil_Kp, hasil_Ki, hasil_Kd;

int en1 = 6; //inisial en1 masuk pin pwm yaitu 6
int en2 = 5; //inisial en2 masuk pin pwm yaitu 5
int in1 = 13; //inisial pin input1/in1 masuk pin 13
int in2 = 12; //inisial pin input2/in2 masuk pin 12
int in3 = 11; //inisial pin input3/in3 masuk pin 11
int in4 = 9; //inisial pin input4/in4 masuk pin 10
volatile intptr_t *portPWM;
byte bitPWM;

float pid_output1, pid_i_mem1, pid_error1,
pid_last_error1, pid_setpoint1 = 35, pid_p1, pid_i1,
pid_d1, pid_max1 = 100;
int data;
float kelembaban;
float suhu;
float kelembaban11, kelembaban21, kelembaban31,
kelembaban41;
float suhu11, suhu21, suhu31, suhu41;
float a, b, c, d;
float x, y, z, w;
byte termometru[8] = //icon for termometer
{
    B00100,
    B01010,
    B01010,
    B01110,
    B01110,
    B11111,
    B11111,
    B01110
};
```

```
byte picatura[8] = //icon for water droplet
{
  B00100,
  B00100,
  B01010,
  B01010,
  B10001,
  B10001,
  B10001,
  B01110,
};

void setup() {
  lcd.begin();

  // Turn on the backlight and print a message.
  lcd.backlight();
  lcd.print("S1 Teknik Elektro");
  lcd.setCursor(1,1);
  lcd.print("Rosida");

  pinMode(en1, OUTPUT); //pin en1 sebagai OUTPUT
  pinMode(en2, OUTPUT); //pin en2 sebagai OUTPUT
  pinMode(in1, OUTPUT); //pin in1 sebagai OUTPUT
  pinMode(in2, OUTPUT); //pin in2 sebagai OUTPUT
  pinMode(in3, OUTPUT); //pin in3 sebagai OUTPUT
  pinMode(in4, OUTPUT); //pin in4 sebagai OUTPUT
  Serial.begin(9600);

  lcd.createChar(1,termometru);
  lcd.createChar(2,picatura);

  dht.begin();
  dht2.begin();
  dht3.begin();
```

```
dht4.begin();
TCCR2A = _BV(WGM21);
TCCR2B = _BV(CS22) | _BV(CS21) | _BV(CS20);
OCR2A = maxDutyCycle;
OCR2B = setDutyCycle(0);
TIMSK2 = _BV(OCIE2B);
*digitalPinToPCMSK(ZCD) |= bit
(digitalPinToPCMSKbit(ZCD));
PCIFR |= bit (digitalPinToPCICRbit(ZCD));
PCICR |= bit (digitalPinToPCICRbit(ZCD));
portPWM = (volatile intptr_t *)
portOutputRegister(digitalPinToPort(PWM));
bitPWM = digitalPinToBitMask(PWM);
dutyCycle = setDutyCycle(0);
pinMode(PWM, OUTPUT);
//turn the PID on
}

void loop() {
  lcd.clear();
  lcd.setCursor (0,1);
  //munculkan icon
  lcd.write(1);
  // Kolom ke 2 baris ke 2
  lcd.setCursor (2,1);
  lcd.print(suhu);
  lcd.print((char)223);
  lcd.setCursor (10,1);
  lcd.write(2);
  lcd.setCursor (12,1);
  lcd.print(kelembaban);

float kelembaban1 = dht.readHumidity();
```



```
float kelembaban2 = dht2.readHumidity();
float kelembaban3 = dht3.readHumidity();
float kelembaban4 = dht4.readHumidity();
float suhu1 = dht.readTemperature();
float suhu2 = dht2.readTemperature();
float suhu3 = dht3.readTemperature();
float suhu4 = dht4.readTemperature();

//rumus kalibrasi DHT22
a = (kelembaban1 + 4.9318)/1.0128;
kelembaban11 = a; //inisialisasi kelembaban1 setelah
kalibrasi
b = (kelembaban2 + 4.9318)/1.0128;
kelembaban21 = b; //inisialisasi kelembaban2 setelah
kalibrasi
c = (kelembaban3 + 4.9318)/1.0128;
kelembaban31 = c; //inisialisasi kelembaban1 setelah
kalibrasi
d = (kelembaban4 + 4.9318)/1.0128;
kelembaban41 = d; //inisialisasi kelembaban2 setelah
kalibrasi

kelembaban = (kelembaban11 + kelembaban21 +
kelembaban31 + kelembaban41 )/ 4;

x = ( suhu1 + 6.5135)/1.2743;
suhu11 = x;
y = ( suhu2 + 6.5135)/1.2743;
suhu21 = y;
z = ( suhu3 + 6.5135)/1.2743;
suhu31 = z;
w = ( suhu4 + 6.5135)/1.2743;
suhu41 = w;
```

```
suhu = (suhu11 + suhu21 + suhu31 + suhu41 )/ 4;

Serial.print("P: ");
Serial.print(hasil_Kp);
Serial.print(" ");
Serial.print("i: ");
Serial.print(hasil_Ki);
Serial.print(" ");
Serial.print("d: ");
Serial.print(hasil_Kd);
Serial.print(" ");
Serial.print("Kelembaban: ");
Serial.println(kelembaban);
Serial.print(" ");
Serial.print("Suhu: ");
Serial.println(suhu);
delay(1000);

Fuzzy_Matrix_Kp();
Fuzzy_Matrix_Ki();
Fuzzy_Matrix_Kd();
Fuzzification();
Defuzzification_Kp();
Defuzzification_Ki();
Defuzzification_Kd();
pid_calculate1();
Serial.print(" delta: ");
Serial.print(pid_last_error1);
Serial.print(" ");
Serial.print(" error: ");
Serial.println(pid_error1);
}
```

```
uint16_t setDutyCycle(uint8_t dutyCycle)
{
    return maxDutyCycle - (1.0 * dutyCycle / 100 *
(maxDutyCycle - 1)) + 1;
}
#if defined(__AVR_ATmega328P__)
# if ((ZCD >= 0) & (ZCD <= 7))
ISR (PCINT2_vect)
# elif ((ZCD >= 8) & (ZCD <= 13))
ISR (PCINT0_vect)
# elif ((ZCD >= 14) & (ZCD <= 19))
ISR (PCINT1_vect)
# endif
#elif defined(__AVR_ATmega2560__)
# if ((ZCD >= 50) & (ZCD <= 53)) || ((ZCD >= 10) & (ZCD
<= 13))
ISR (PCINT0_vect)
# elif ((ZCD >= 14) & (ZCD <= 15))
ISR (PCINT1_vect)
# elif ((ZCD >= A8) & (ZCD <= A15))
ISR (PCINT2_vect)
# endif
#endif
{
    OCR2B = dutyCycle;
    TCNT2 = 0;
}
ISR (TIMER2_COMPB_vect)
{
    *portPWM |= bitPWM;
    delayMicroseconds(30);
    *portPWM &= ~bitPWM;
```

```
}  
void kipas(int s1, int s2) {  
    analogWrite(en1, s1);  
    digitalWrite(in1, LOW);  
    digitalWrite(in2, HIGH);  
    analogWrite(en2, s2);  
    digitalWrite(in3, HIGH);  
    digitalWrite(in4, LOW);  
  
}  
void pid_calculate1() {  
    pid_error1 = suhu - pid_setpoint1;  
  
    pid_p1 = hasil_Kp;  
    pid_i1 = hasil_Ki;  
    pid_d1 = hasil_Kd,  
  
    pid_i_mem1 += pid_i1 * pid_error1;  
    if (pid_i_mem1 > pid_max1)pid_i_mem1 = pid_max1;  
    else if (pid_i_mem1 < pid_max1 * -1)pid_i_mem1 =  
pid_max1 * -1;  
  
    pid_output1 = pid_p1 * pid_error1 + pid_i_mem1 + pid_d1  
* (pid_error1 - pid_last_error1);  
    if (pid_output1 > pid_max1)pid_output1 = pid_max1;  
    else if (pid_output1 < pid_max1 * -1)pid_output1 =  
pid_max1 * -1;  
  
    pid_last_error1 = pid_error1;  
  
    data = abs(pid_output1);
```

```
if (pid_error1>0){
    kipas(255,255);
    dutyCycle = setDutyCycle(0);
}
else{
    dutyCycle = setDutyCycle(data);
    kipas(0,0);
}

void Fuzzy_Matrix_Kp(){
    Fuzzy_Kp[0][0]=min(NE,NDE);
    Fuzzy_Kp[0][1]=min(ZE,NDE);
    Fuzzy_Kp[0][2]=min(PE,NDE);
    Fuzzy_Kp[1][0]=min(NE,ZDE);
    Fuzzy_Kp[1][1]=min(ZE,ZDE);
    Fuzzy_Kp[1][2]=min(PE,ZDE);
    Fuzzy_Kp[2][0]=min(NE,PDE);
    Fuzzy_Kp[2][1]=min(ZE,PDE);
    Fuzzy_Kp[2][2]=min(PE,PDE);
}

void Fuzzy_Matrix_Ki(){
    Fuzzy_Ki[0][0]=min(NE,NDE);
    Fuzzy_Ki[0][1]=min(ZE,NDE);
    Fuzzy_Ki[0][2]=min(PE,NDE);
    Fuzzy_Ki[1][0]=min(NE,ZDE);
    Fuzzy_Ki[1][1]=min(ZE,ZDE);
    Fuzzy_Ki[1][2]=min(PE,ZDE);
    Fuzzy_Ki[2][0]=min(NE,PDE);
    Fuzzy_Ki[2][1]=min(ZE,PDE);
    Fuzzy_Ki[2][2]=min(PE,PDE);
}

void Fuzzy_Matrix_Kd(){
```

```
Fuzzy_Kd[0][0]=min(NE,NDE);
Fuzzy_Kd[0][1]=min(ZE,NDE);
Fuzzy_Kd[0][2]=min(PE,NDE);
Fuzzy_Kd[1][0]=min(NE,ZDE);
Fuzzy_Kd[1][1]=min(ZE,ZDE);
Fuzzy_Kd[1][2]=min(PE,ZDE);
Fuzzy_Kd[2][0]=min(NE,PDE);
Fuzzy_Kd[2][1]=min(ZE,PDE);
Fuzzy_Kd[2][2]=min(PE,PDE);
}
void Fuzzification(){
    NE=0; ZE=0; PE=0;
    NDE=0; ZDE=0; PDE=0;

    if(pid_error1<=-5) NE=1;
    else if(pid_error1>=5) PE=1;
    else{
        if(pid_error1>=-5){
            NE =((-5) - pid_error1)/(-5);
            ZE=1-NE;
        }
        else{
            PE=(pid_error1/5);
            ZE=1-PE;
        }
    }
}
if(pid_last_error1<=-3) NDE=1;
else if(pid_last_error1>=3) PDE=1;
else{
    if(pid_last_error1>=-3){
        NDE =((-3) - pid_last_error1)/(-3);
        ZDE=1-NDE;
    }
}
```

```
else{
    PDE=(pid_last_error1/3);
    ZDE=1-PDE;
}
}
}

void Defuzzification_Kp(){
    SKp=max(Fuzzy_Kp[0][0],max(Fuzzy_Kp[1][0],
max(Fuzzy_Kp[2][0], Fuzzy_Kp[1][1])));
    MKp=max(Fuzzy_Kp[0][1],Fuzzy_Kp[2][1]);

    BKp=max(Fuzzy_Kp[0][2],max(Fuzzy_Kp[1][2],Fuzzy_Kp[2][2]
));
    if(!(SKp==0 && MKp==0 && BKp==0)){
        hasil_Kp=(SKp*OSKp+MKp*OMKp+BKp*OBKp)/(SKp+MKp+BKp);
    }
}

void Defuzzification_Ki(){

    SKi=max(Fuzzy_Ki[0][2],max(Fuzzy_Ki[1][2],Fuzzy_Ki[2][2]
));

    MKi=max(Fuzzy_Ki[0][1],max(Fuzzy_Ki[1][1],Fuzzy_Ki[2][1]
));

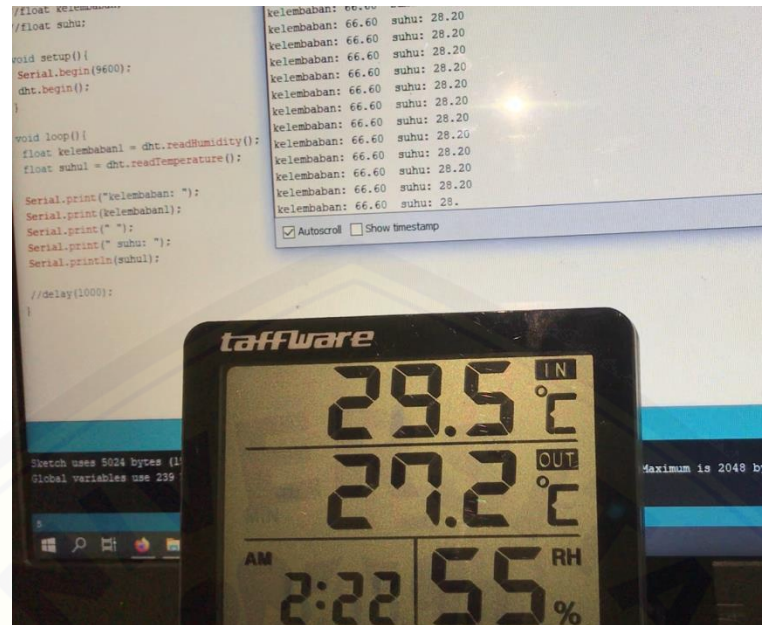
    BKi=max(Fuzzy_Ki[0][0],max(Fuzzy_Ki[1][0],Fuzzy_Ki[2][0]
));
    if(!(SKi==0 && MKi==0 && BKi==0)){
        hasil_Ki=(SKi*OSKi+MKi*OMKi+BKi*OBKi)/(SKi+MKi+BKi);
    }
}
```

```
void Defuzzification_Kd(){  
  
SKd=max(Fuzzy_Kd[0][0],max(Fuzzy_Kd[1][0],max(Fuzzy_Kd[  
2][1],Fuzzy_Kd[1][1])));  
  
MKd=max(Fuzzy_Kd[0][1],max(Fuzzy_Kd[2][1],Fuzzy_Kd[1][2  
]));  
BKd=max(Fuzzy_Kd[0][2],Fuzzy_Kd[2][2]);  
if(!(SKd==0 && MKd==0 && BKd==0)){  
hasil_Kd=(SKd*OSKd+MKd*OMKd+BKd*OBKd)/(SKd+MKd+BKd);  
}  
}
```

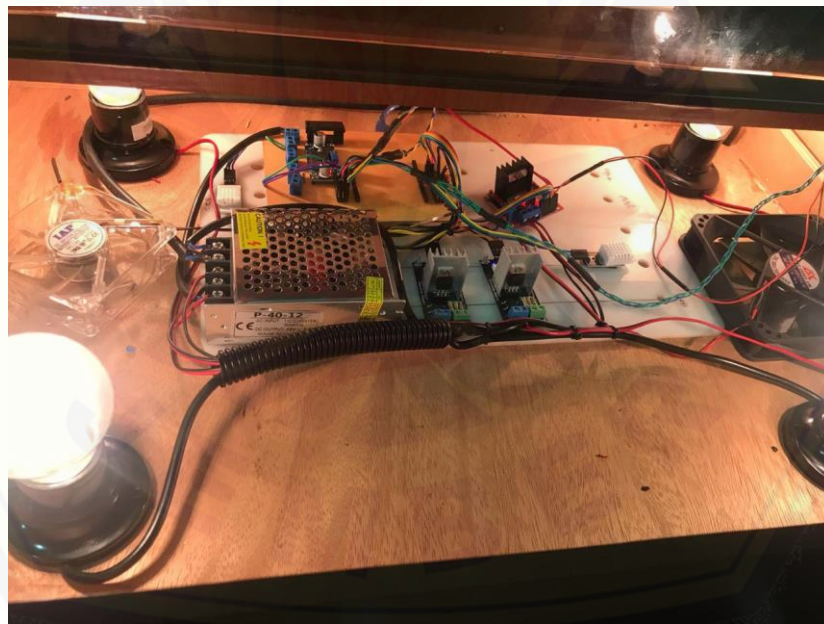




## 2. Gambar Hardware



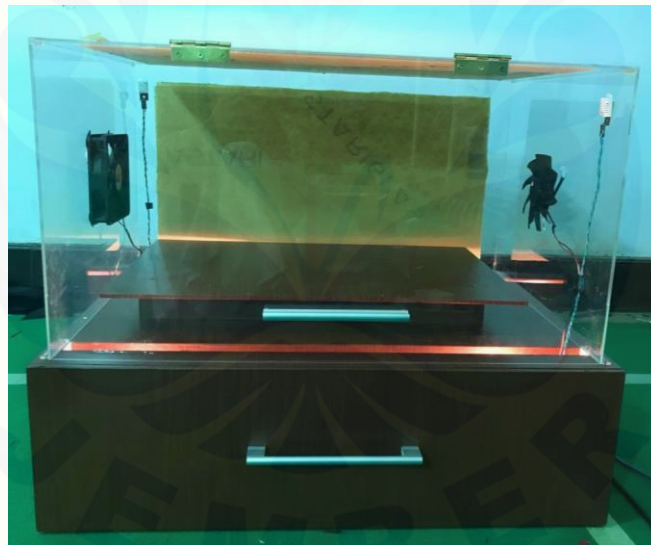
Gambar Kalibrasi Sensor DHT22



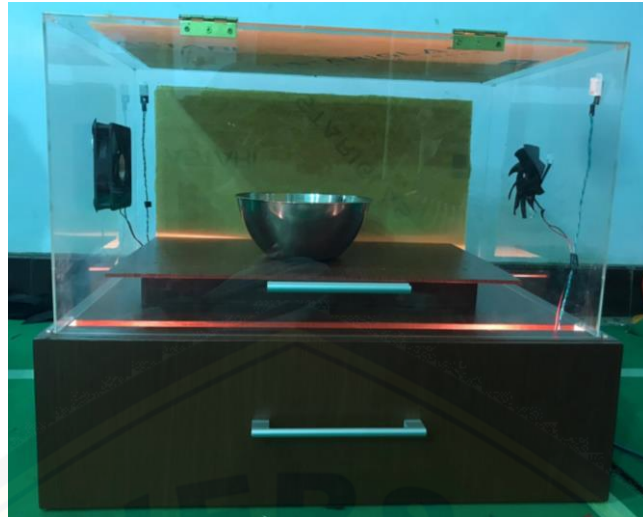
Gambar rangkaian kontroler, modul AC *light dimmer* dengan lampu pijar, modul *driver* L298N dengan kipas.



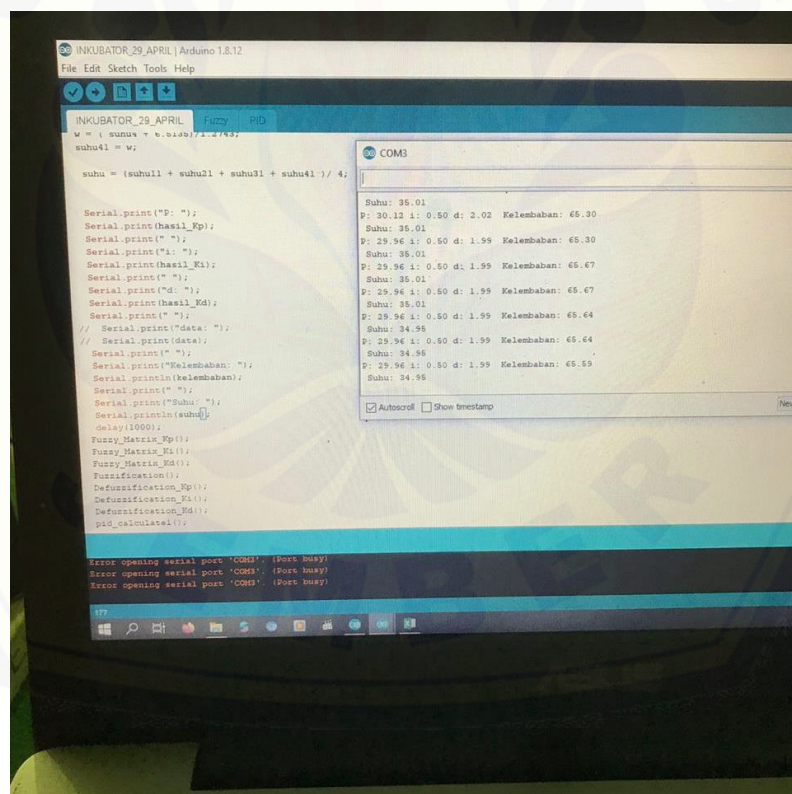
Gambar *Box* Bagian Atas Inkubator



Gambar Pengujian Inkubator Tanpa Beban Secara Keseluruhan dengan Metode *Fuzzy-PID*



Gambar Pengujian Inkubator dengan Beban Secara Keseluruhan dengan Metode *Fuzzy*-PID



Gambar Pengambilan Data

### 3. Biaya Pengeluaran Pembuatan Inkubator Bayi

Tabel Biaya Pengeluaran Pembuatan Inkubator Bayi

No	Pembelian	Jumlah	Harga Satuan	Jumlah Harga
1.	Akrilik 3mm	1 lembar (1,5m x 1m)	450.000	450.000
2.	Arduino Uno	1 buah	85.000	85.000
3.	Modul AC <i>Light Dimmer</i>	2 buah	80.000	160.000
4.	Driver L298N	1 buah	25.000	25.000
5.	Kipas Aksial	4 buah	16.000	64.000
6.	Adaptor Power Supply	1 buah	65.000	65.000
7.	Sensor DHT22	4 buah	45.000	180.000
8.	Lampu 60 Watt	4 buah	8.500	34.000
9.	<i>Cutting</i> Akrilik	-	225.000	225.000
10.	Box Bawah & Tatakan Bayi	1 buah	275.000	275.000
11.	Kabel USB Type B 3M	1 buah	22.000	22.000
12.	LCD 16x2	1 buah	20.000	20.000
13.	Kabel Jumper	1 set (40 buah)	15.000	15.000
			Total	1.620.000