

**RANCANG BANGUN MESIN PENETAS TELUR MENGGUNAKAN
KONTROL LOGIKA FUZZY DAN KONTROL ON-OFF**

SKRIPSI
TEKNIK ELEKTRO KONSENTRASI TEKNIK KONTROL

Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik



PATRICK SIMBOLON
NIM. 115060305111007

UNIVERSITAS BRAWIJAYA
FAKULTAS TEKNIK
JURUSAN TEKNIK ELEKTRO
MALANG
2016

LEMBAR PENGESAHAN

RANCANG BANGUN MESIN PENETAS TELUR MENGGUNAKAN KONTROL LOGIKA FUZZY DAN KONTROL ON-OFF

SKRIPSI

TEKNIK ELEKTRO KONSENTRASI TEKNIK KONTROL

Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik



Disusun oleh :
PATRICK SIMBOLON
NIM. 115060305111007

Skripsi ini telah direvisi dan disetujui oleh dosen pembimbing Pada
tanggal 12 Agustus 2016

Dosen Pembimbing I

Dosen Pembimbing II

Muhammad Aziz Muslim, S.T., M.T., Ph.D.

Ir. Retnowati, M.T.

NIP. 19741203 200012 1 001

NIP. 19511224 198203 2 001

Mengetahui,

Ketua Jurusan Teknik Elektro

Muhammad Aziz Muslim, S.T., M.T., Ph.D.

NIP. 19741203 200012 1 001

JUDUL SKRIPSI:

RANCANG BANGUN MESIN PENETAS TELUR MENGGUNAKAN KONTROL LOGIKA FUZZY DAN KONTROL ON-OFF.

Nama Mahasiswa : PATRICK SIMBOLON

NIM : 115060305111007-63

Program Studi : TEKNIK ELEKTRO

Konsentrasi : TEKNIK KONTROL

Komisi Pembimbing :

Ketua : M.Aziz Muslim, S.T., M.T., Ph.D

Anggota : Ir. Retnowati, M.T.

TIM DOSEN PENGUJI :

Dosen Penguji 1 : Ir. Purwanto, M.T.

Dosen Penguji 2 : Dr.Ir. Bambang Siswojo, M.T.

Dosen Penguji 3 : Dr. Ir. Erni Yudaningtyas, M.T.

Tanggal Ujian : 12 AGUSTUS 2016

SK Penguji : No. 979 /UN10.6/SK/2016

PERNYATAAN ORISINALITAS SKRIPSI

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya dan berdasarkan hasil penelusuran berbagai karya ilmiah, gagasan dan masalah ilmiah yang diteliti dan diulas di dalam Naskah Skripsi ini adalah asli dari pemikiran saya. Tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu Perguruan Tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata di dalam naskah Skripsi ini dapat dibuktikan terdapat unsur-unsur jiplakan, saya bersedia Skripsi dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, pasal 25 ayat 2 dan pasal 70).

Malang, 12 Agustus 2016

Mahasiswa,

Materai Rp 6000,-

Patrick Simbolon

NIM. 115060305111007



PENGANTAR

Puji syukur kepada Tuhan Yesus Kristus atas hikmat, bimbingan, dan kuasa-Nya sehingga penulisan skripsi dengan judul “Rancang Bangun Mesin Penetas Telur Menggunakan Kontrol Logika *Fuzzy* dan Kontrol *On-Off*” dapat diselesaikan dengan baik dan tepat pada waktunya.

Penulis menyadari bahwa penyusunan skripsi ini tidak lepas dari bantuan, bimbingan serta dorongan dari berbagai pihak. Pada kesempatan ini penulis menyampaikan rasa terima kasih yang sebesar – besarnya kepada:

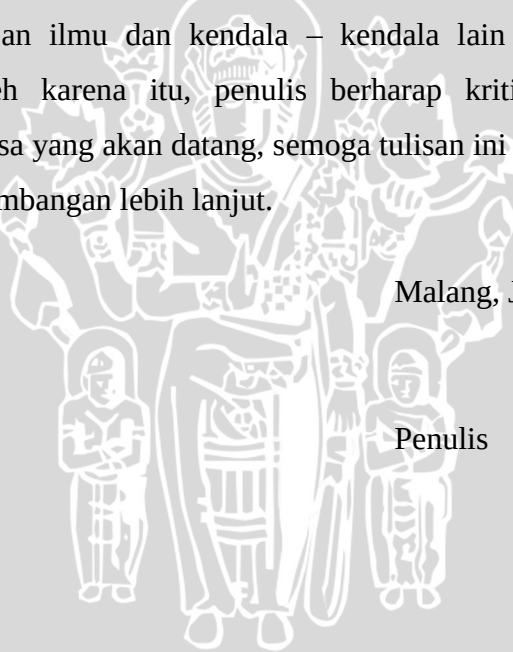
- Kedua orang tua Raja Simbolon dan Miati, dan adik Monica Patricia Simbolon, Ricardo Andika Simbolon, Natalia Desti Simbolon telah banyak memberikan doa, kasih sayang, dukungan, serta semangat.
- Bapak M. Aziz Muslim, S.T., M.T., Ph.D. selaku Ketua Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya Malang dan dosen pembimbing skripsi yang telah memberikan banyak waktu dan tenaganya untuk membimbing dari awal, memberikan saran, nasehat-nasehat, dan pelajaran..
- Bapak Ir. Purwanto, M.T. selaku Ketua Kelompok Dosen Keahlian Teknik Kontrol Teknik Elektro, Fakultas Teknik Universitas Brawijaya dan dosen pembimbing skripsi yang telah memberikan banyak waktu dan tenaganya untuk membimbing dari awal, memberikan saran, nasehat-nasehat, dan pelajaran.
- Bapak Hadi Suyono S.T., M.T., Ph.D selaku Sekretaris Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya.
- Bapak M. Ali Mustofa, S.T., M.T. selaku Ketua Program Studi Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya yang banyak memberikan pengarahan dalam hal akademik dan penulisan skripsi.
- **Bapak Ir. Unggul Wibawa, M.Sc.** Selaku dosen penasehat akademik. Yang telah banyak memberikan nasihat-nasihat akademiknya.
- Bapak, Ibu dosen serta segenap staf dan karyawan Jurusan Teknik Elektro baik secara langsung maupun tidak langsung yang telah banyak membantu dalam menyelesaikan skripsi ini.
- Saudara-saudara Inverter 2011, KMK Teknik, dan semua rekan-rekan di Jurusan Teknik Elektro dan Fakultas Teknik Universitas Brawijaya.

- Muhammad Zainudin, S.T. dan Desta Aran Putra, S.T. selaku pembimbing ketiga dalam penelitian atas segala bimbingan, nasihat, arahan, motivasi, saran dan masukannya.
- Bude Anik di Blitar yang turut mendukung dan memotivasi dalam pembuatan skripsi ini.
- Seluruh rekan-rekan asisten Laboratorium Sistem Kontrol yang telah memfasilitasi penelitian ini.
- Bapak tukang kayu yang telah membuat kotak penetas telur.
- Teman-teman serta semua pihak yang tidak mungkin dicantumkan namanya satu persatu telah memberikan bantuan serta dukungan baik secara langsung maupun tidak langsung atas penyusunan skripsi ini.

Dalam penyusunan skripsi ini, penulis menyadari bahwa skripsi ini belum sempurna, karena keterbatasan ilmu dan kendala – kendala lain yang terjadi selama pengerjaan skripsi ini. Oleh karena itu, penulis berharap kritik dan saran untuk penyempurnaan tulisan di masa yang akan datang, semoga tulisan ini dapat bermanfaat dan dapat digunakan untuk pengembangan lebih lanjut.

Malang, Juli 2016

Penulis



DAFTAR ISI

RINGKASAN	i
SUMMARY	ii
PENGANTAR	iii
DAFTAR ISI	v
DAFTAR TABEL	viii
DAFTAR GAMBAR	ix
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan	3
1.5 Sistematika Pembahasan	3
BAB II TINJAUAN PUSTAKA	4
2.1 Teori Dasar Penetasan Telur Menggunakan Mesin Tetas	4
2.2 Teori Dasar Kebutuhan Suhu dan Kelembaban untuk Telur Tetas	4
2.3 Sensor Suhu dan Kelembaban	5
2.4 Pemanas(Heater)	6
2.5 Humidifier	7
2.6 Driver Motor L298N	7
2.7 Rangkaian Dimmer	8
2.7.1 Rangkaian TRIAC	8
2.7.2 Rangkaian Zero Cross Detector	9
2.8 Arduino Mega	9
2.9 Catu Daya	10
2.10 PWM	11
2.11 Kontrol Logika Fuzzy	12
2.11.1 Fungsi Keanggotaan Fuzzy	12
2.11.2 Sistem Inferensi Fuzzy	13
2.11.3 Rule Base Fuzzy	14
2.11.4 Defuzzyfikasi	15

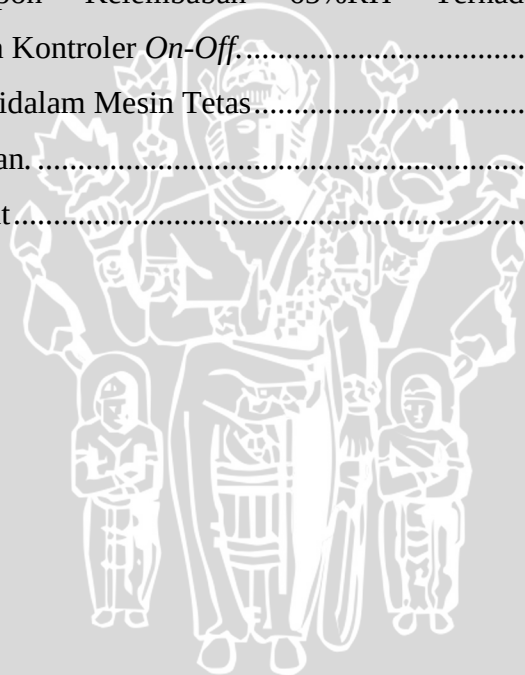
BAB III METODOLOGI	17
3.1 Studi Literatur	17
3.2 Studi Komparasi	17
3.3 Penentuan Spesifikasi Alat	17
3.4 Perancangan Sistem	18
3.5 Realisasi Pembuatan Sistem	18
3.6 Pengujian dan Analisis Data	18
3.7 Pengambilan Kesimpulan.....	18
 BAB IV PERANCANGAN DAN PEMBUATAN ALAT	 19
4.1 Perancangan Sistem	19
4.2 Diagram Blok Sistem	19
4.3 Spesifikasi Alat	20
4.4 Prinsip Kerja	21
4.5 Perancangan Perangkat Keras.....	21
4.5.1 Pemilihan Pemanas.....	21
4.5.2 Rangkaian <i>Dimmer</i>	22
4.5.3 Pemodelan Sensor SHT11.....	23
4.6 Perancangan Algoritma Kontrol Logika <i>Fuzzy</i>	23
4.6.1 Perancangan Fungsi Keanggotaan.....	23
4.6.2 Perancangan <i>Rules Base</i>	25
4.7 Modul Arduino Mega 2560	26
4.8 <i>Flowchart</i> Perancangan Perangkat Lunak	28
4.8.1 <i>Flowchart</i> Perancangan Perangkat Lunak Suhu.....	28
4.8.2 <i>Flowchart</i> Perancangan Perangkat Lunak Kelembaban.....	29
 BAB V PENGUJIAN DAN ANALISIS SISTEM	 30
5.1 Pengujian Sensor Suhu dan Kelembaban SHT11.....	30
5.1.1 Peralatan yang Digunakan	30
5.1.2 Prosedur Pengujian	30
5.1.3 Hasil Pengujian.....	31
5.2 Pengujian Pengujian Sensor Kelembaban SHT11	32
5.2.1 Peralatan yang Digunakan.....	32

5.2.2	Prosedur Pengujian.....	32
5.2.3	Hasil Pengujian.....	33
5.3	Pengujian Penyulutan TRIAC.....	34
5.3.1	Peralatan yang Digunakan	34
5.3.2	Prosedur Pengujian	34
5.3.3	Hasil Pengujian.....	35
5.4	Pengujian Rangkaian <i>Dimmer</i>	40
5.4.1	Peralatan yang Digunakan.....	40
5.4.2	Prosedur Pengujian.....	40
5.4.3	Hasil Pengujian.....	41
5.5	Pengujian Keseluruhan Sistem.....	43
5.5.1	Peralatan yang Digunakan.....	43
5.5.2	Prosedur Pengujian.....	43
5.5.3	Hasil Pengujian.....	44
5.5	Hasil Penetasan.....	52
BAB VI PENUTUP		55
6.1	Kesimpulan.....	55
6.2	Saran.....	55
DAFTAR PUSTAKA		56
LAMPIRAN		57

DAFTAR GAMBAR

Gambar 2.1	Telur Ayam Fertil	5
Gambar 2.2	Sensor SHT11	6
Gambar 2.3	Rangkaian Sensor SHT11	6
Gambar 2.4	Lampu Pijar 60 Watt	6
Gambar 2.5	<i>Humidifier USB</i>	7
Gambar 2.6	<i>Driver Motor LN298N</i>	7
Gambar 2.7	Kipas DC 12V	8
Gambar 2.8	<i>AC Dimmer Module</i>	8
Gambar 2.9	Arduino Mega	10
Gambar 2.10	Pengaturan Tegangan AC dengan PWM	11
Gambar 2.11	Himpunan Fungsi Keanggotaan Segitiga.....	12
Gambar 2.12	Nilai keanggotaan untuk himpunan <i>fuzzy</i> segitiga.....	13
Gambar 2.13	Fungsi Implikasi <i>Min</i>	13
Gambar 2.14	Fungsi Implikasi <i>Dot</i>	14
Gambar 4.1	Diagram Blok <i>Plant</i> Mesin Tetas	19
Gambar 4.2	Pemanas pada Mesin Tetas	22
Gambar 4.3	Skema Rangkaian <i>Dimmer</i>	22
Gambar 4.4	Modul Rangkaian <i>Dimmer</i>	23
Gambar 4.5	Perancangan <i>membership function error</i> suhu	24
Gambar 4.6	Perancangan <i>membership function delta error</i> suhu	24
Gambar 4.7	Perancangan <i>membership function</i> untuk PWM.....	25
Gambar 4.8	Modul Arduino Mega 2560	26
Gambar 4.9	<i>Flowchart</i> Sistem Suhu.....	28
Gambar 4.10	<i>Flowchart</i> Sistem Kelembaban.....	29
Gambar 5.1	Blok Pengujian Sensor Suhu SHT11	30
Gambar 5.2	Perbandingan Suhu Sensor SHT11 terhadap Termometer Digital	32
Gambar 5.3	Blok Pengujian Sensor SHT11 untuk Kelembaban	32
Gambar 5.4	Grafik Pengujian Sensor Kelembaban SHT11	34
Gambar 5.5	Rangkaian Pengujian Sudut Penyalaan TRIAC.....	35
Gambar 5.6	Bentuk Gelombang Keluaran Arduino	35
Gambar 5.7	Bentuk Gelombang Keluaran TRIAC dengan PWM 32.....	36
Gambar 5.8	Grafik Hubungan Nilai PWM dengan Tegangan.....	41

Gambar 5.9	Karakteristik Plant	42
Gambar 5.10	Sistem Keseluruhan	43
Gambar 5.11	Grafik Respon Sistem Suhu dengan Kontrol Logika <i>Fuzzy</i>	44
Gambar 5.12	Grafik Respon Sistem Kelembaban 60%RH dengan kontroler <i>On-Off</i>	45
Gambar 5.13	Grafik Respon Sistem Kelembaban 65%RH dengan kontroler <i>On-Off</i>	47
Gambar 5.14	Grafik Respon Sistem Suhu Terhadap Gangguan Menggunakan Kontrol Logika <i>Fuzzy</i>	48
Gambar 5.15	Respon Sistem Kelembaban 60%RH Terhadap Gangguan Menggunakan Kontroler <i>On-Off</i>	49
Gambar 5.16	Grafik Respon Kelembaban 65%RH Terhadap Gangguan Menggunakan Kontroler <i>On-Off</i>	51
Gambar 5.17	Telur Tetas Didalam Mesin Tetas.....	52
Gambar 5.18	Hasil Penetasan.....	53
Gambar 5.19	Menetas Cacat.....	53



DAFTAR TABEL

Tabel 2.1	Suhu dan Kelembaban untuk Telur Ayam	5
Tabel 4.1	Perancangan <i>Rule Base Fuzzy</i> Pengontrol Suhu	25
Tabel 4.2	Penggunaan Pin Pada Arduino Mega 2560	26
Tabel 5.1	Hasil Pengujian Sensor Suhu SHT11	31
Tabel 5.2	Hasil Pengujian Sensor Kelembaban SHT11	33
Tabel 5.3	Hasil Pengujian Sudut Penyalan TRIAC	36
Tabel 5.4	Kesalahan Pengujian PWM Penyalan TRIAC	40
Tabel 5.5	Perbandingan Nilai PWM Terhadap Tegangan	41
Tabel 5.6	Perbandingan Nilai PWM Terhadap <i>steady state</i> Suhu	42
Tabel 5.7	Data Respon Suhu Menggunakan Kontrol Logika <i>Fuzzy</i>	45
Tabel 5.8	Data Respon Kelembaban 60%RH Menggunakan Kontroler <i>On-Off</i>	46
Tabel 5.9	Data Respon Kelembaban 65%RH Menggunakan Kontroler <i>On-Off</i>	47
Tabel 5.10	Data Respon Suhu Terhadap Gangguan Menggunakan Kontrol Logika <i>Fuzzy</i>	48
Tabel 5.11	Data Respon Kelembaban 60%RH Terhadap Gangguan Menggunakan Kontroler <i>On-Off</i>	50
Tabel 5.12	Data Respon Kelembaban 60%RH Terhadap Gangguan Menggunakan Kontroler <i>On-Off</i>	51
Tabel 5.13	Hasil Penetasan	53

RINGKASAN

Patrick Simbolon, Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya, Maret 2016, *Rancang Bangun Mesin Penetas Telur Menggunakan Kontrol Logika Fuzzy dan Kontrol On-Off* Dosen Pembimbing : Muhammad Aziz Muslim, Retnowati.

Masalah yang sering dihadapi dalam proses penetasan telur ayam adalah kebutuhan suhu dan kelembaban di dalam mesin tetas harus sesuai dengan kebutuhan telur tetas, sehingga dibutuhkan sistem pengendalian suhu dan kelembaban yang dapat disesuaikan dengan kebutuhan telur tetas. Suhu dan kelembaban yang diaplikasikan di dalam mesin tetas untuk telur ayam kampung dengan suhu $37,5^{\circ}\text{C}$ dan kelembaban 55-65%RH. Dimana kelembaban dibagi menjadi 2 yaitu, 55-60%RH pada hari ke 1-18 dan 65%RH pada hari ke 19-21.

Pada penelitian ini proses pengendalian suhu di dalam mesin tetas menggunakan Kontrol Logika Fuzzy dengan 3 *Membership Function* (MF) menggunakan metode Inferensi *Max-Min* dan metode Defuzzifikasi *Weighted Average*. Penetapan *rule base fuzzy* sebanyak 25 *rule*. Untuk *input* yang digunakan yaitu *error* dan *delta error* dengan rentang -0,5 sampai 0,5, sedangkan untuk *output* yaitu nilai PWM dengan rentang 10 sampai 90.

Hasil pengujian pada suhu awal $28,39^{\circ}\text{C}$ dengan *setpoint* suhu $37,5^{\circ}\text{C}$ diperoleh *error steady state* sebesar 0,21% dengan *settling time* sebesar 690 detik dan kelembaban dengan *set point* 60%RH pada hari ke 11-18 diperoleh *error steady state* sebesar 0,64% dan *settling time* sebesar 560 detik sedangkan pada hari ke 19-21 dengan *setpoint* 65%RH diperoleh *error steady state* sebesar 1,08%. Saat diberi gangguan sistem juga mampu menjaga suhu dan kelembaban yang sesuai dengan *setpoint*.

Dari hasil pengujian keseluruhan maka sistem sudah memenuhi kebutuhan suhu dan kelembaban yang dibutuhkan selama proses penetasan. Proses penetasan telur selama 21 hari dan diperoleh hasil dari 18 telur yang ditetaskan, 16 telur menetas dan 2 telur tidak menetas. Maka diperoleh persentase penetasan sebesar 88,89. Dari hasil pengujian dan penetasan tersebut sudah memenuhi tujuan dari skripsi ini yaitu membuat mesin tetas dengan meningkatkan respon sistem suhu dan kelembaban yang lebih baik dari mesin tetas yang telah dibuat sebelumnya dan persentase penetasan melebihi 80%.

Kata kunci : Telur Tetas Ayam Kampung, Mesin Tetas, Suhu dan Kelembaban, Kontrol Logika Fuzzy, Kontrol On-Off.

SUMMARY

Patrick Simbolon, Department of Electrical Engineering, Faculty of Engineering of Brawijaya Univerdity, March 2016, *Designing and Making of Egg Incubator Using Fuzzy Logic Control and On-Off*, Advisors: Muhammad Aziz Muslim, Retnowati.

The problem which usually faced in the chicken egg hatchery process is the needs of temperature and humidity inside of egg Incubator must suitable with the hatch eggs so it is required temperature and humidity controlling system that is adjustable with the needs of hatch eggs. Temperature and humidity which were applied in this incubator for Domestic Chicken with the temperature is $37,5^{\circ}\text{C}$ and the humidity stays in the range of 55-65%RH. The humidity itself divided into two parts, 55-65%RH on 1st - 18th day and 65%RH on 19th - 21st day.

In this research, the temperature controlling process in the Incubator is using Fuzzy Logic Control with 3 Membership Function (MF) that used Max-Min Inferention method and Weighted Average Defuzzification method. Determination of rule based fuzzy was 25 rules. For the *input* that is used were error and delta error with the range from -0,5 to 0,5 , meanwhile for the *output* was PWM point with the range starts from 10 to 90. From all the results, the system has been fulfilled the needs of temperature and humidity which were required during hatchery process. The hatchery process is 21 days and the result was from 18 eggs which were hatched, there were 16 eggs success to hatch and 2 eggs not success to hatch. Then, the hatchery percentage is about 88,89%.

Keyword : Domestic Chicken Hatch Eggs, Hatching Machine, Temperature and Humidity, Fuzzy Logic Control, On-Off Control

BAB I PENDAHULUAN

1.1 Latar Belakang

Dunia peternakan khususnya bidang perunggasan dituntut untuk menghasilkan produk peternakan yang kualitas dan kuantitasnya harus tinggi, baik itu dari daging maupun telur. Untuk memenuhinya maka harus tersedia unggas dalam jumlah yang banyak pula. Jika hanya bergantung secara alami, maka jumlah unggas yang dibutuhkan tidak tercapai, oleh karena itu diperlukan teknologi yang dalam waktu yang singkat dapat menghasilkan anakan unggas dalam jumlah yang banyak. Hal ini dapat dilakukan dengan penetasan telur memakai mesin tetas.

Mesin penetas atau inkubator ada yang berbentuk sederhana, semi otomatis dan modern. Alat ini dapat menetas telur dalam jumlah banyak dan memiliki kapasitas yang bervariasi yaitu dari 100 sampai 10.000 butir. Dengan menggunakan mesin tetas dapat mendorong industri perunggasan dalam penyediaan bibit unggul dalam jumlah besar dan dalam waktu yang bersamaan. Faktor-faktor yang mempengaruhi daya tetas perlu diketahui untuk mengukur tingkat keberhasilan suatu penetasan.

Pada prinsipnya penetasan telur dengan mesin tetas adalah mengkondisikan telur sama seperti telur yang dierami oleh induknya. Baik itu suhu, kelembaban dan juga posisi telur. Dalam proses penetasan dengan menggunakan mesin tetas memiliki kelebihan dibanding dengan penetasan secara alami, yaitu : dapat dilakukan sewaktu-waktu, dapat dilakukan dengan jumlah telur yang banyak, menghasilkan anak dalam jumlah banyak dalam waktu bersamaan, dapat dilakukan pengawasan dan seleksi pada telur (Yuwanta, 1983).

Penetasan buatan sama dengan penetasan alami, yaitu menyediakan kondisi lingkungan (temperatur, kelembaban, dan sirkulasi udara) yang sesuai agar embrio dalam telur berkembang dengan optimal, sehingga telur dapat menetas (Sukardi, 1999). Penetasan dengan alat tetas buatan terbagi atas dua cara, yaitu dengan matahari dan sekam serta mesin tetas. Alat – alat ini sederhana, bahkan dapat kita buat sendiri. Dari kedua jenis ini pun terdapat bermacam- macam jenis alat tetas yang prinsip kerjanya sama, karena umumnya menggunakan tenaga panas, baik panas matahari maupun panas listrik atau lampu teplok (Paimin, 2000).

Hal-hal yang mendukung keberhasilan dari penetasan dengan mesin tetas antara lain adalah telur tetas itu sendiri harus telur yang fertil yaitu telur dari betina yang di kawini

pejantan. Suhu dalam mesin tetas sekitar $37,5^{\circ}\text{C}$, kelembaban 60%RH pada hari ke 1-18 dan 65%RH pada hari ke 19-21 (Riyanto, 2001), sirkulasi udara dalam mesin tetas lancar, dan juga candling yaitu peneropongan telur selama proses penetasan sehingga dapat diketahui pertumbuhan embrionya.

Dengan membandingkan hasil respon dan persentase penetasan dari skripsi sebelumnya (Galih Hatiyanto, Teknik Elektro Universitas Brawijaya, Tahun Lulus : 2011) mesin tetas ini harus memiliki respon suhu dan kelembaban yang lebih baik. Untuk metode yang dapat digunakan adalah pengaturan suhu pada mesin tetas menggunakan Kontrol Logika *Fuzzy* dan pengaturan kelembaban menggunakan kontroler *On-Off*. Dengan adanya Kontrol Logika *Fuzzy* dan kontroler *On-Off* pada mesin tetas diharapkan suhu dan kelembaban sesuai kebutuhan telur tetas.

1.2 Rumusan Masalah

Berdasarkan latar belakang dapat disusun rumusan masalah sebagai berikut :

- Bagaimana merancang perangkat keras dan perangkat lunak Kontrol Logika *Fuzzy* untuk pengendali suhu dan kontroler *On-Off* sebagai pengendali kelembaban berbasis Arduino Mega 2560 dalam penetasan telur ayam?

1.3 Batasan Masalah

- Mesin tetas yang digunakan adalah *prototype* dengan desain sendiri berdimensi 70cm x 40cm x 50cm.
- Sensor suhu dan kelembaban yang digunakan adalah sensor SHT11.
- Mikrokontroler yang digunakan adalah Arduino Mega 2560.
- Set point suhu $37,5^{\circ}\text{C}$ dan set point kelembaban 60%RH pada hari ke 1-18, 65% pada hari ke 19-21.
- Jumlah telur ayam kampung yang ditetaskan yaitu 18 butir.
- Pemanas yang digunakan yaitu 2 buah lampu pijar 60Watt dengan *dimmer* sebagai pengendali nyala terang redup lampu (Kontrol Logika *Fuzzy*).
- Media penambah kelembaban yaitu *humidifier* (Kontrol *On-Off*).
- Kipas DC 12V digunakan untuk menurunkan suhu dan sirkulasi udara (Kontrol *On-Off*).
- Motor AC 220V sebagai pemutar rak telur tidak dibahas secara mendalam.

1.4 Tujuan

- Membuat mesin tetas yang mampu mengatur suhu dan kelembaban yang sesuai dengan kebutuhan telur ayam tetas.
- Menghasilkan respon sistem dan persentase penetasan yang lebih baik dari skripsi sebelumnya (Galih Hatiyanto, Teknik Elektro Universitas Brawijaya, Tahun Lulus : 2011).

1.5 Sistematika Pembahasan

BAB I Pendahuluan

Bab ini berisi tentang uraian latar belakang, tujuan, batasan masalah, rumusan masalah, manfaat serta sistematika penulisan.

BAB II Tinjauan Pustaka

Tinjauan Pustaka berisi dasar teori penunjang penelitian. Pustaka yang diambil adalah pustaka yang relevan dan sesuai serta mendukung penelitian, seperti buku-buku budidaya ayam kampung, teori Kontrol Logika *Fuzzy*, sensor suhu dan kelembaban, identifikasi sistem, Arduino dan lain-lain. Selain dari buku pustaka juga akan diambil dari jurnal, internet, dan sumber pengetahuan yang lain.

BAB III Metodologi Penelitian

Bab ini berisi tentang metode yang digunakan dalam pengerjaan alat seperti studi literatur, studi komparasi, perancangan dan pembuatan perangkat keras dan perangkat lunak, pengujian alat, pengambilan data dan analisis data yang digunakan dalam skripsi ini.

BAB IV Perancangan dan Pembuatan

Bab ini menjelaskan tentang perancangan dan pembuatan alat yang meliputi *block diagram* sistem, prinsip kerja alat, perancangan perangkat keras dan perangkat lunak.

BAB V Pengujian dan Analisis Sistem

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

BAB VI Kesimpulan dan Saran

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

BAB II

TINJAUAN PUSTAKA

2.1 Teori Dasar Penetasan Telur Menggunakan Mesin Tetas

Penetasan telur adalah usaha untuk menetasakan telur unggas dengan bantuan mesin penetasan telur yang sistem atau cara kerjanya mengadopsi tingkah laku (behaviour) induk ayam atau unggas lainnya selama masa mengeram. Perbanyakkan populasi unggas biasanya ditempuh dengan cara menetasakan telur yang sudah dibuahi. Menurut Paimin (2000) pentasan telur ada dua cara, yaitu melalui penetasan alami (induk ayam) dan melalui penetasan buatan (mesin tetas). Kapasitas produksi unggas sekali pengeraman hanya sekitar 10 – 15 butir telur. Akan tetapi, untuk mesin tetas sangat bervariasi tergantung kapasitas mesinnya.

Mesin tetas adalah ruangan tertutup yg dipanasi dengan aliran listrik atau pemanas buatan lainnya yang dipakai untuk mengerami dan menetasakan telur. Pengeraman dengan mesin tetas dilakukan oleh peternak biasanya karena telur yang ditetaskan relatif banyak. Peternak yang bermodal besar biasanya lebih memilih menggunakan mesin tetas karena lebih efektif dan efisien. Biasanya mesin tetas telur dilengkapi dengan pemanas, pemutar telur, dan sensor suhu dan kelembaban sehingga suhu dan kelembaban ruangan ideal bagi telur tetas.

2.2 Teori Dasar Kebutuhan Suhu dan Kelembaban untuk Telur Tetas

Telur-telur yang dihasilkan oleh induk unggas tidak seluruhnya berkualitas baik. Untuk itu ada beberapa kriteria sehingga dikatakan telur tetas itu baik untuk ditetaskan diantaranya :

- Bentuk Telur

Bentuk telur yang baik berbentuk normal yaitu telur yang berbentuk sedikit agak lonjong. Bagian atas agak besar dan bawahnya lebih kecil dan tumpul.

- Keadaan Kulit Telur

Kulit yang permukannya halus dan merata.

- Umur Telur

Umur telur yang ditetaskan sebaiknya telur yang umumnya dibawah dari 7 hari.

Kebutuhan temperatur pada mesin tetas dapat diberikan dengan bantuan pemanas modern atau konvensional. Pemanas konvensional biasanya hanya menggunakan lampu minyak. Berikut Tabel 2.1 kebutuhan suhu dan kelembaban yang ideal untuk mentaskan telur ayam dan Gambar 2.1 contoh telur fertil.

Tabel 2.1 Suhu dan Kelembaban untuk Telur Ayam

Umur (hari)	Suhu (°C)	Kelembaban (%)
1 - 18	37,5	55 - 60
19 - 21	37,5	65

(Sumber: Farry Paimin, 2000)



Gambar 2.1 Telur Ayam Fertil

(Sumber: Anonim, 2016)

2.3 Sensor Suhu dan Kelembaban

SHT11 Module merupakan modul sensor suhu dan kelembaban relatif. Modul ini dapat digunakan sebagai alat pengindra suhu dan kelembaban dalam aplikasi pengendali suhu dan kelembaban ruangan maupun aplikasi pemantau suhu dan kelembaban relatif ruangan. Sensor SHT11 ini adalah sensor suhu dan kelembaban relatif berbasis sensirion SHT11.

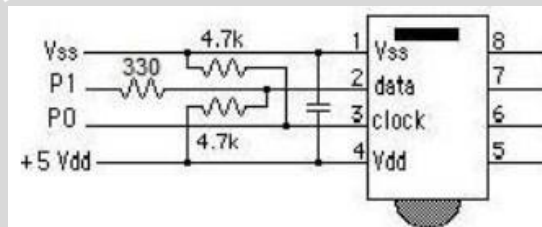
Sensor SHT11 dapat mengukur suhu dari -40°C hingga $123,8^{\circ}\text{C}$ atau dari -40°F hingga $4,9^{\circ}\text{F}$ dan kelembaban relatif dari 0%RH hingga 1%RH. Memiliki ketepatan (akurasi) pengukuran suhu hingga $0,5^{\circ}\text{C}$ pada suhu $^{\circ}\text{C}$ dan ketepatan (akurasi) pengukuran kelembaban relatif hingga 3,5%RH. Memiliki antarmuka serial *synchronous 2-wire*, bukan I2C. Jalur antarmuka telah dilengkapi dengan rangkaian pencegah kondisi sensor *lock-up*. Sensor SHT11 membutuhkan catu daya +5V DC dengan konsumsi daya rendah $30\mu\text{W}$ dan memiliki faktor bentuk 8 pin DIP 0,6 sehingga memudahkan pemasangannya.

Pin yang digunakan pada sensor ini ada dua pin yaitu pin SCK dan pin DATA. Pin SCK berfungsi untuk mensinkronisasikan komunikasi antara sensor dan Arduino, sedangkan pin DATA berfungsi untuk mentransfer data in dan out pada sensor. Dibawah ini Gambar 2.2

menunjukkan sensor SHT11 dan Gambar 2.3 menunjukkan rangkaian sensor suhu dan kelembaban (SHT11).



Gambar 2.2 Sensor SHT11
(Sumber: Sensirion.com)



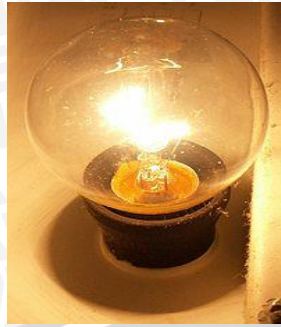
Gambar 2.3 Rangkaian Sensor SHT11
(Sumber: www.sensirion.com)

2.4 Pemanas (Heater)

Pemanas yang digunakan dalam perancangan alat ini berupa 2 buah lampu pijar AC 60W. Dengan adanya rangkaian dimmer lampu pijar dapat menyesuaikan tingkat cahaya sesuai dengan suhu yang dibutuhkan telur ayam kampung.

Persyaratan elemen pemanas antara lain :

- Harus tahan lama pada suhu yang dikehendaki.
- Sifat mekanisnya harus kuat pada suhu yang dikehendaki.
- Koefisien muai harus kecil, sehingga perubahan bentuknya pada suhu yang dikehendaki tidak terlalu besar.



Gambar 2.4 Lampu Pijar 60Watt
(Sumber : www.wikipedia.org)

2.5 *Humidifier*

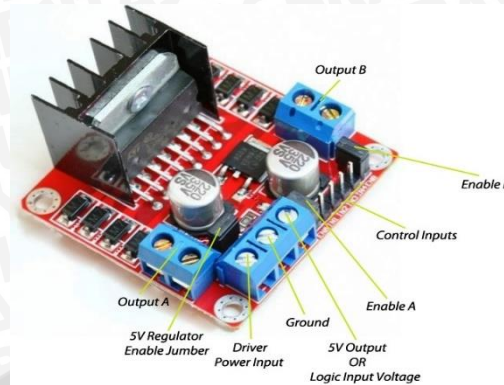
Humidifier berfungsi sebagai penambah kelembaban pada ruang mesin tetas dengan cara menyemburkan kabut dari air melalui proses kavitasi. Aktuator yang digunakan dalam perancangan alat berupa alat penambah kelembaban ruangan yang terdapat dipasaran. Driver yang digunakan untuk menjalankan *humidifier* berupa relay yang aktif ketika diberi tegangan 5V. *Humidifier* diharapkan dapat menjaga kelembaban dengan sesuai kebutuhan telur ayam kampung. (lihat Gambar 2.5)



Gambar 2.5 *Humidifier USB*
(Sumber: www.tokopedia.com)

2.6 *Driver Motor L298N*

Driver motor L298N seperti ditunjukkan Gambar 2.6 digunakan untuk mengendalikan putaran motor DC yang menjadi penggerak kipas DC 12V seperti Gambar 2.7 sebagai media untuk menurunkan suhu pada mesin tetas. Modul ini dihubungkan dengan *output* dari mikrokontroler Arduino Mega 2560.



Gambar 2.6 Driver Motor L298N

(Sumber: electrosome.com)

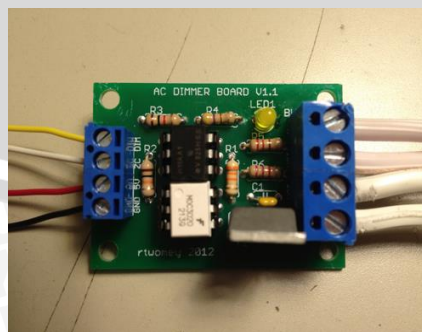


Gambar 2.7 Kipas DC 12V

(Sumber: www.globalindustries.com)

2.7 Rangkaian Dimmer

Diperlukan sebuah rangkaian yang bisa digunakan untuk melakukan pengendalian AC, rangkaian tersebut rangkaian *dimmer* yang di dalamnya terdiri dari rangkaian TRIAC dan rangkaian *Zero Cross Detector*. Gambar 2.8 menunjukkan contoh modul rangkaian *dimmer* AC.



Gambar 2.8 AC Dimmer Module

2.7.1 Rangkaian TRIAC

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

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

2.7.2 Rangkaian Zero Cross Detector

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

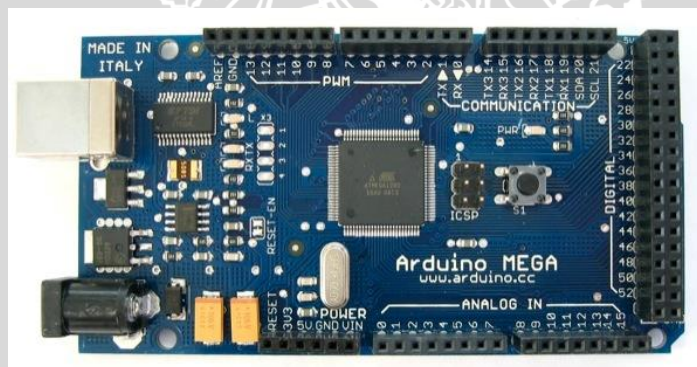
Pada rangkaian *zero cross detector* dibutuhkan rangkaian yang bisa mendeteksi persimpangan antara gelombang tegangan bolak-balik dengan titik nol agar pengaturan fasa dapat dilakukan pada titik nol gelombang tegangan bolak-balik. Metode ini berfungsi untuk menentukan frekuensi suatu gelombang dengan cara mendeteksi banyaknya *zero point* pada suatu rentang waktu. *Zero crossing detector* berfungsi untuk mendeteksi perpotongan gelombang sinus pada tegangan AC dengan *zero point* tegangan AC tersebut, sehingga dapat memberikan sinyal acuan saat dimulainya pemicuan sinyal *Pulse Width Modulation* (PWM). Dengan rangkaian *zero crossing detector*, bisa dilakukan pendeteksian *zero point* sekaligus

mengubah suatu sinyal sinusoidal menjadi sinyal kotak. Sinyal keluaran rangkaian *zero cross detector* ini akan dimasukkan ke mikrokontroler, oleh karena itu dibutuhkan juga komponen yang mampu memisahkan tegangan 5V dan 220V. Untuk rangkaian ini digunakan komponen optocoupler 4N25.

2.8 Arduino Mega

Arduino Mega adalah *board* mikrokontroler berbasis Atmega1280. Memiliki 54 pin *input* dan *output* digital, 14 pin diantara pin tersebut dapat digunakan sebagai *output* PWM dan 16 pin *input* analog, 16 MHz osilator kristal, koneksi USB, jack power, ICSP header, dan tombol reset. Untuk mendukung mikrokontroler agar dapat digunakan, cukup hanya menghubungkan *Board* Arduino Mega ke komputer dengan menggunakan kabel USB atau dengan adaptor AC-DC untuk menjalankannya.

Secara fisik, ukuran Arduino Mega hampir kurang lebih 2 kali lebih besar dari Arduino Uno, ini untuk mengakomodasi lebih banyaknya pin Digital dan Analog pada *board* Arduino Mega tersebut. Tampilan Arduino Mega dapat dilihat pada Gambar 2.9.



Gambar 2.9 Arduino Mega

(Sumber: <http://arduino.cc/en/Main/ArduinoBoardMega>)

2.9 Catu Daya

Arduino mega dapat diaktifkan melalui koneksi USB atau dengan catu daya eksternal. Sumber catu daya akan dipilih secara otomatis. Eksternal (non-USB) catu daya dapat berasal baik dari adaptor AC-DC atau baterai. Adaptor ini dapat dihubungkan dengan menancapkan sebuah 2.1mm konektor pusat-positif ke konektor listrik pada *board*. Baterai dapat dimasukkan ke dalam *header* pin Gnd dan Vin dari konektor catu daya.

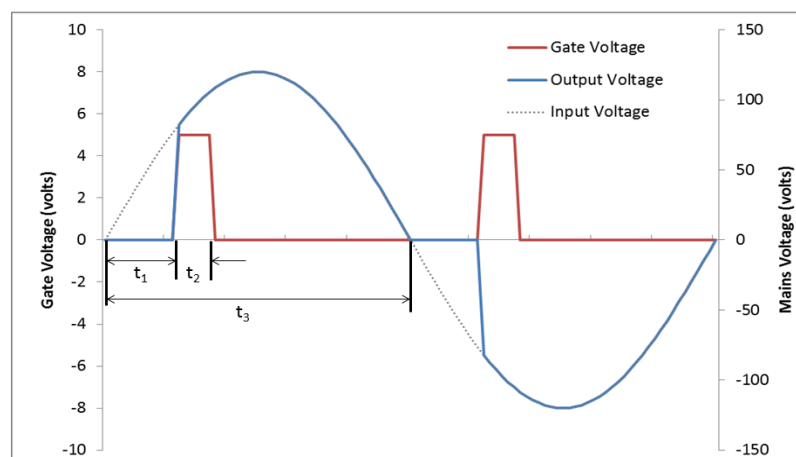
Board dapat beroperasi pada pasokan eksternal 7 sampai 12 volt. apabila dicatu kurang dari 7V, pin 5V dapat menyediakan kurang dari 5V dan *board* bisa menjadi tidak stabil. Jika menggunakan lebih dari 12V, regulator tegangan bisa panas dan merusak *board*. Kisaran yang disarankan adalah 7 sampai 12 volt.

Pin untuk catu daya adalah sebagai berikut:

- a. VIN: Adalah *intput* tegangan untuk *board* Arduino ketika menggunakan sumber catu daya eksternal.
- b. 5V: Sebuah pin yang mengeluarkan tegangan ter-regulator 5 Volt. Dari pin ini tegangan sudah diatur (ter-regulator) dari regulator yang tersedia (*built-in*) pada *board*. Arduino dapat diaktifkan dengan sumber catu daya baik berasal dari *jack power* DC (7-12 Volt), konektor USB (5 Volt), atau pin VIN pada *board* (7-12 Volt). Memberikan tegangan melalui pin 5V atau 3.3V secara langsung tanpa melewati regulator dapat merusak *board* Arduino.
- c. 3.3V: Sebuah pin yang menghasilkan tegangan 3,3 Volt. Tegangan ini dihasilkan oleh regulator yang terdapat pada papan (*on-board*). Arus maksimum yang dihasilkan adalah 50 mA.
- d. GND: Pin *Ground*.

2.10 PWM (*Pulse Width Modulation*)

PWM (*Pulse Width Modulation*) digunakan untuk mengatur tegangan yang masuk pada heater. Gambar 2.10 menunjukkan sinyal PWM (*Pulse Width Modulation*) untuk mengatur tegangan AC yang masuk pada *heater*.



Gambar 2.10 Pengaturan Tegangan AC dengan PWM

(Sumber: <http://playground.arduino.cc>)

Dalam Gambar 2.10, setelah persimpangan nol gelombang terdeteksi, triac akan *off* dalam waktu (t_1). Saat triac *off* maka tegangan AC juga masih *off*. Setelah triac aktif (ditampilkan dalam warna merah), gelombang AC juga akan keluar hingga saat gelombang melewati persimpangan nol. Walaupun triac sudah tidak aktif (t_2), gelombang AC akan tetap aktif sampai persimpangan nol. Hal ini akan terus berulang saat gelombang melewati persimpangan nol yang selanjutnya (t_3).

2.11 Kontrol Logika Fuzzy

2.11.1 Fungsi Keanggotaan Fuzzy

Fungsi keanggotaan ($\mu_f = \text{membership function}$) merupakan suatu kurva yang menunjukkan pemetaan antara titik-titik *input* data ke dalam nilai keanggotaannya (derajat keanggotaan).

Ambil U sebagai semesta dari himpunan objek $\{u\}$. Himpunan fuzzy F dalam semesta pembicaraan U dinyatakan dalam nilai keanggotaan μ_f yang mempunyai interval nilai $[0,1]$. Himpunan fuzzy biasanya dinyatakan dengan :

$$F = \{ \langle u, \mu_f(u) \rangle \mid u \in U \}$$

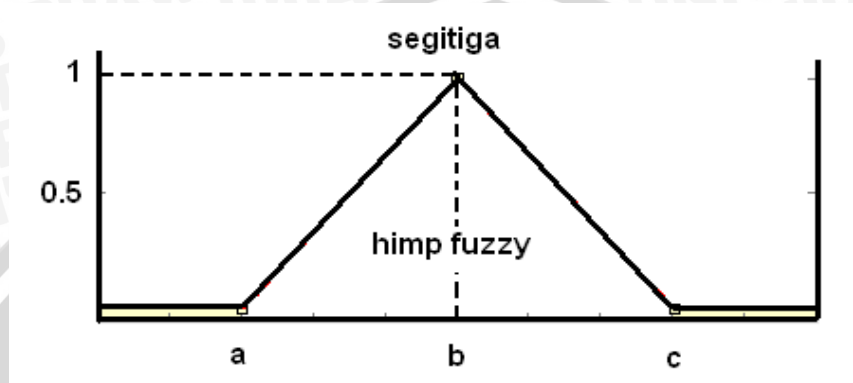
jika U kontinyu maka himpunan F dapat ditulis dengan :

$$F = \int \mu_f(u)/u$$

jika U diskrit maka himpunan F dapat ditulis dengan :

$$F = \sum \mu_f(u_i)/u_i$$

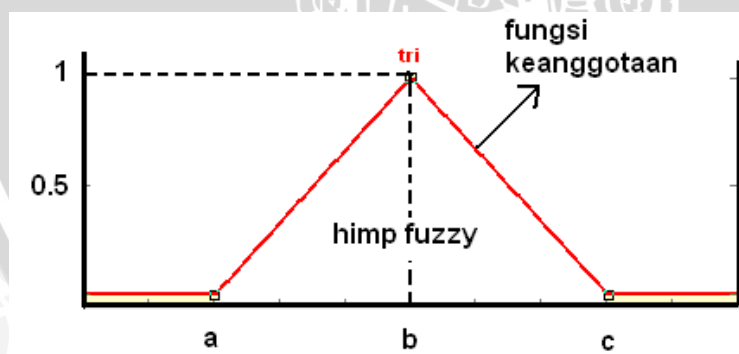
Fungsi Keanggotaan Segitiga (Gambar 2.11) :



Gambar 2.11 Himpunan Fungsi Keanggotaan Segitiga

$$S(u; a, b, c) = \begin{cases} (u - a)/(b - a) & \text{untuk } a \leq u \leq b \\ (c - u)/(c - b) & \text{untuk } b \leq u \leq c \\ 0 & \text{untuk } u > c \end{cases}$$

Contoh nilai keanggotaan dari suatu himpunan fuzzy dapat dilihat pada Gambar 2.12.



Gambar 2.12 Nilai keanggotaan untuk himpunan fuzzy segitiga

2.11.2 Sistem Inferensi Fuzzy

Pada umumnya tiap-tiap aturan (proposisi) fuzzy dinyatakan dalam bentuk IF..THEN.. dan menyatakan suatu hubungan tertentu. Hubungan fuzzy ini sering disebut implikasi. Hubungan fuzzy dalam *knowledge base* dapat didefinisikan sebagai

himpunan implikasi fuzzy. Ada 2 jenis proposisi fuzzy yaitu “*condition fuzzy proposition*” dan “*uncondition fuzzy proposition*”.

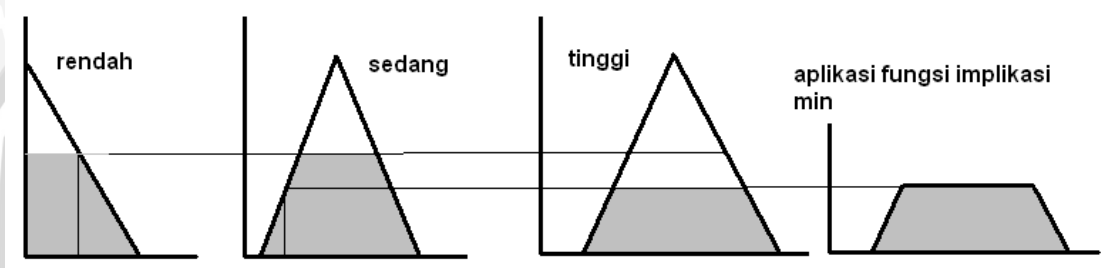
1. *Condition Fuzzy Proposition*

Jenis ini dicirikan dengan penggunaan IF.

IF x is A THEN y is B

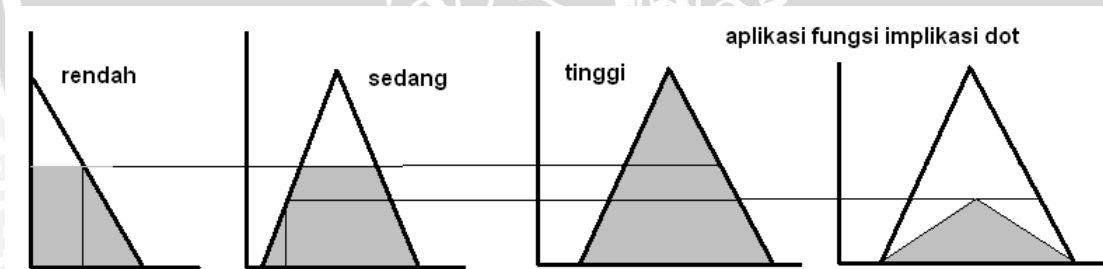
Jika suatu proposisi menggunakan bentuk terkondisi maka ada 2 fungsi implikasi yang dapat digunakan yaitu :

- Minimum : fungsi ini memotong *output* himpunan fuzzy. (lihat Gambar 2.13)



Gambar 2.13 Fungsi Implikasi *Min*

- Dot : fungsi ini menskalakan *output* himpunan fuzzy. (lihat Gambar 2.14)



Gambar 2.14 Fungsi Implikasi *Dot*

2. *Uncondition Fuzzy Proposition.*

Jenis *uncondition* ditandai dengan tidak adanya pernyataan IF.

x is A

proposisi *uncondition* selalu diaplikasi dengan model AND.

Jika dalam system fuzzy terdapat beberapa aturan, maka ada 3 metode yang dipakai dalam menentukan inferensi yaitu : *max-min*, *additive* dan *probabilistic OR* (*probor*).

1. Metode *Max-Min*

Max dapat dianalogikan dengan operasi logika OR sedangkan Min dianalogikan dengan operasi logika AND.

2. Metode *additive*

Metode *additive* dilakukan dengan melakukan *bounded-sum* terhadap semua *output* daerah fuzzy. Secara umum dapat dituliskan dengan L:

$$\mu_{sf}[x_i] \leftarrow \min (1, \mu_{sf}[x_i] + \mu_{kf}[x_i])$$

$\mu_{sf}[x_i]$ = nilai keanggotaan solusi fuzzy sampai aturan ke-i

$\mu_{kf}[x_i]$ = nilai keanggotaan konsekuen (*output*) fuzzy sampai aturan ke-i

2.11.3 *Rules Base Fuzzy*

Basis aturan terdiri dari sejumlah aturan yang biasanya dinyatakan secara *linguistic*. Aturan fuzzy seringkali dinyatakan dengan “IF...THEN....”. Hal ini didasarkan pada kebutuhan akan :

1. Memfasilitasi kepakaran (*expert*) manusia dengan cara yang tepat untuk mengekspresikan pengetahuan dan pengalaman mereka.
2. Memfasilitasi desainer dengan cara yang mudah untuk membuat dan memprogram aturan fuzzy.
3. Mengurangi biaya desain.
4. Meningkatkan efisiensi pengambilan keputusan fuzzy (*fuzzy inference*).

Jika variabel *input* lebih dari satu, biasanya ada penghubung yang menyatakan relasi dari tiap-tiap *input*. Penghubung itu biasanya dinyatakan dengan ‘AND’, ‘OR’, dan ‘ALSO’. Pemakaian penghubung itu misalnya:

IF x_1 IS A_{k1} OR x_2 IS A_{k2} THEN y_1 IS B_{k1} ALSO y_2 IS B_{k2}

secara lebih singkat dapat dinyatakan:

IF A_{k1} OR A_{k2} THEN B_{k1} ALSO B_k

2.11.4 Defuzzyfikasi

Ada beberapa metode defuzzifikasi antara lain :

1. Metode *Centroid*

Metode centroid dilakukan dengan mengambil titik pusat daerah fuzzy.

Secara umum dirumuskan dengan :

$$z = \frac{\int z\mu(z)dz}{\int \mu(z)dz} \text{ atau } z = \frac{\sum_{j=1}^n z_j \mu(z_j)}{\sum_{j=1}^n \mu(z_j)},$$

2. Metode *Bisector*

Metode bisector dilakukan dengan mengambil nilai dari domain fuzzy yang memiliki nilai keanggotaan separo dari nilai keanggotaan fuzzy. Secara dirumuskan dengan :

$$z_p \text{ sedemikian hingga } \int_{R1}^p \mu(z)dz = \int_p^{Rn} \mu(z)dz$$

3. Metode *Mean of Maximum* (MOM)

Metode MOM dilakukan dengan mengambil nilai rata-rata domain yang memiliki nilai keanggotaan maksimum.

4. Metode *Largest of Minimum* (LOM)

Metode LOM dilakukan dengan mengambil nilai terbesar dari domain yang memiliki nilai keanggotaan maksimum.

5. Metode *Smallest of Maximum* (SOM)

Metode LOM dilakukan dengan mengambil nilai terkecil dari domain yang memiliki nilai keanggotaan maksimum

BAB III

METODOLOGI

Untuk merealisasikan alat yang telah dirancang, langkah – langkah yang dilakukan adalah sebagai berikut:

3.1 Studi Literatur

Studi literatur fokus dilakukan pada prinsip kerja alat yang dirancang meliputi sistem pengendalian suhu dan kelembaban ruangan mesin tetas, karakteristik dari komponen yang digunakan serta studi tentang penggunaan perangkat Arduino Mega sebagai perangkat akuisisi data.

Adapun hal-hal yang berhubungan dengan hal tersebut adalah:

- a. Studi tentang penetasan telur ayam serta suhu dan kelembaban yang dibutuhkan.
- b. Studi tentang spesifikasi sistem yang digunakan dan rangkaian elektrik pendukung lainnya.

3.2 Studi Komparasi

Studi komparasi dilakukan untuk membandingkan pengujian dan hasil analisis alat yang telah dibuat dengan alat yang telah dibuat sebelumnya (Galih Hatiyanto, Teknik Elektro Universitas Brawijaya, Tahun Lulus: 2011). Studi komparasi ini bertujuan untuk mengembangkan respon dari sistem dan hasil persentase yang lebih baik.

3.3 Penentuan Spesifikasi Alat

Adapun spesifikasi alat sebagai berikut:

- a. Mesin tetas, merupakan tempat berlangsungnya proses pengendalian suhu dan kelembaban telur tetas dengan dimensi 70cm x 40cm x 50cm.
- b. Lampu pijar AC 60W sebagai elemen pemanas.
- c. Kipas DC 12V akan membantu proses penurunan suhu di dalam mesin tetas.
- d. *Humidifier*, berfungsi sebagai penambah kelembaban di dalam mesin tetas dengan cara menyemburkan uap dari proses kativasi.
- e. Sensor SHT11, untuk membaca suhu dan kelembaban di dalam mesin tetas.
- f. Penyalan terang redup lampu AC dikendalikan menggunakan *dimmer*.
- g. *Relay* digunakan sebagai *driver* untuk *humidifier*.
- h. *Driver motor* L298N digunakan sebagai *driver* untuk Kipas DC 12V
- i. *Dimmer* menggunakan Kontrol Logika Fuzzy.
- j. Kipas DC 12V dan *humidifier* menggunakan kontroler *On-Off*.

3.4 Perancangan Sistem

Perancangan sistem dilakukan sebagai langkah awal sebelum terbentuknya suatu sistem beserta rangkaian elektronik serta pemrogramannya, hal ini dimaksudkan agar proses pembuatan mesin tetas yang sesuai dengan kebutuhan telur ayam dapat berjalan sesuai deskripsi awal yang telah direncanakan. Perancangan sistem yang dilakukan meliputi:

- a. Penentuan dimensi dan bahan yang digunakan dalam pembuatan mesin tetas.
- b. Perancangan elektronik.
- c. Perancangan program pada perangkat Arduino Mega sebagai media akuisisi data untuk proses indentifikasi *plant*.
- d. Blok diagram sistem.

3.5 Realisasi Pembuatan Sistem

Untuk merealisasikan alat atau sistem yang telah dirancang secara keseluruhan, maka perlu diperhatikan beberapa hal dalam pembuatan alat tersebut, diantaranya yaitu:

- a. Pembuatan purwarupa mesin tetas secara keseluruhan yang meliputi pemilihan *heater*, sensor, kipas, *humidifier* dan komponen lainnya.
- b. Pembuatan rangkaian elektronik yang meliputi rangkaian *dimmer*, *driver*, *relay*, LCD dan pengendali sinyal.
- c. Pengisian program yang telah dirancang sebelumnya pada perangkat Arduino Mega 2560.

3.6 Pengujian dan Analisis Data

Setelah semua komponen pada alat sudah terhubung sesuai dengan diagram blok sistem yang telah dirancang dan perangkat lunak untuk mendukung sistem yang telah dibuat, maka diadakan pengujian dan analisa alat. Pengujian dilakukan secara per blok terlebih dahulu kemudian dilanjutkan secara keseluruhan sistem.

3.7 Pengambilan Kesimpulan

Kesimpulan diambil berdasarkan data yang diperoleh dari pengujian sistem secara keseluruhan. Jika hasil yang didapatkan telah sesuai dengan yang direncanakan sebelumnya, maka proses tersebut telah berhasil memenuhi harapan dan tentunya memerlukan pengembangan lebih lanjut untuk penyempurnaan.

BAB IV

PERANCANGAN DAN PEMBUATAN ALAT

Dalam bab ini akan dibahas mengenai perancangan dan pembuatan sistem pengendalian suhu dan kelembaban mesin tetas. Pada dasarnya perancangan alat meliputi perancangan perangkat keras dan perangkat lunak.

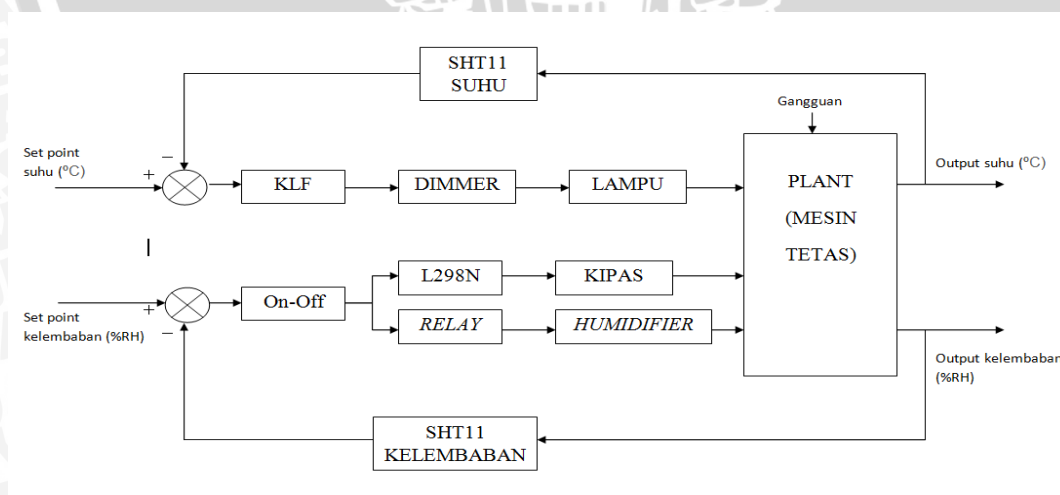
4.1 Perancangan Sistem

Perancangan alat ini dilakukan secara bertahap dalam bentuk blok sehingga akan memudahkan dalam analisis pada setiap bloknnya maupun secara keseluruhan. Perancangan ini terdiri atas:

- Perancangan perangkat keras terdiri atas perancangan sensor suhu dan kelembaban SHT11, 2 kipas DC 12V, 2 lampu pijar AC 60W, 2 *humidifier* USB 5V dan rangkaian *dimmer*.
- Perancangan perangkat lunak Arduino untuk sensor suhu dan kelembaban dan sinyal PWM untuk *driver* pemanas, kipas, *relay humidifier*.

4.2 Diagram Blok Sistem

Sistem umum diagram blok dari identifikasi sistem terdiri dari blok pembangkitan sinyal uji, pencatatan sinyal gangguan dan pencatatan sinyal *output*. Pengambilan data sistem dilaksanakan dengan mengondisikan sistem dengan rangkaian *loop* terbuka, seperti ditunjukkan Gambar 4.1.



Gambar 4.1 Diagram Blok Plant Mesin Tetas

Berdasarkan blok diagram yang ditunjukkan pada Gambar 4.1 dapat dijelaskan sebagai berikut:

- a. *Setpoint* pada sistem untuk suhu adalah $37,5^{\circ}\text{C}$ dan untuk kelembaban adalah 55-65%RH.
- b. Sensor suhu dan kelembaban yang digunakan adalah SHT11 yang berfungsi mendeteksi perubahan suhu dan kelembaban didalam mesin tetas. Perubahan yang terdeteksi oleh sensor nantinya akan menjadi *input* untuk mikrokontroler Arduino Mega 2560.
- c. Mikrokontroler yang digunakan adalah Arduino Mega 2560 yang berfungsi untuk memberikan sinyal PWM (*Pulse Width Modulation*) pada *dimmer*, driver motor L298N dan rangkaian *relay* untuk *humidifier*.
- d. Aktuator untuk pengendali suhu menggunakan lampu pijar sebagai elemen pemanas yang dikendalikan menggunakan Kontrol Logika *Fuzzy*, kipas sebagai penurun suhu dan *humidifier* sebagai penambah kelembaban yang dikendalikan menggunakan kontroler *On-Off*.

4.3 Spesifikasi Alat

Spesifikasi sistem pengendalian suhu dan kelembaban pada mesin tetas menggunakan Kontrol Logika *Fuzzy* dan Kontroler *On-Off* berbasis Arduino Mega 2560 adalah sebagai berikut:

- a. Mesin tetas atau tempat penetasan telur berupa kayu triplek dengan dimensi 70cm x 40cm x 50cm.
- b. Aktuator yang mempengaruhi suhu dan kelembaban pada mesin tetas antara lain lampu pijar AC 60W, kipas DC 12V, dan *humidifier*.
- c. Proses penetasan telur dengan menggunakan 2 buah lampu pijar AC 60W.
- d. Proses untuk menjaga aliran udara dengan menggunakan 2 buah kipas DC 12V yang di pasang pada bagian kiri dan kanan mesin tetas.
- e. Proses untuk menjaga kelembaban agar stabil menggunakan *humidifier* USB 5V yang diletakkan di dalam mesin tetas.
- f. Sensor yang digunakan adalah sensor suhu dan kelembaban SHT11.
- g. Kriteria mesin tetas disesuaikan untuk telur tetas yaitu pada hari ke 1-18 kelembaban 55-60% sedangkan untuk hari ke 19-21 kelembaban 65%.

4.4 Prinsip Kerja

Prinsip kerja sistem penetasan telur sebagai berikut:

- Catu daya yang digunakan ada 6 macam yaitu *power supply* 12V AC, 5V DC, untuk mencatu Arduino Mega 2560, 12V DC untuk mencatu kipas, 5V DC untuk mencatu *humidifier*, 5V untuk mencatu *driver* motor L298N dan 220V AC untuk mencatu rangkaian *dimmer*.
- Sensor SHT11 membaca suhu dan kelembaban pada *plant* (mesin tetas) kemudian masukan dari sensor suhu akan diproses oleh Arduino Mega 2560 dengan algoritma *Fuzzy*. Kemudian Arduino mengirim *output* PWM ke rangkaian *dimmer* yang dapat mengatur intensitas cahaya pada lampu pijar yang berfungsi sebagai media pemanas pada mesin tetas.
- Kipas untuk menurunkan dan mengalirkan udara panas yang berlebih dan *humidifier* sebagai media penambah kelembaban karena semakin panas suhu pada mesin tetas maka kelembaban akan turun, sehingga dibutuhkan alat untuk menambah kelembaban ruang mesin tetas. *Relay* sebagai driver untuk *humidifier* dan *driver motor* L298N sebagai driver untuk kipas menggunakan kontroler *On-Off*.

4.5 Perancangan Perangkat Keras

Perancangan perangkat keras terbagi beberapa bagian, diantaranya:

- Pemilihan elemen panas.
- Perancangan rangkaian *dimmer*
- Permodelan Sensor SHT11.

4.5.1 Pemilihan Pemanas

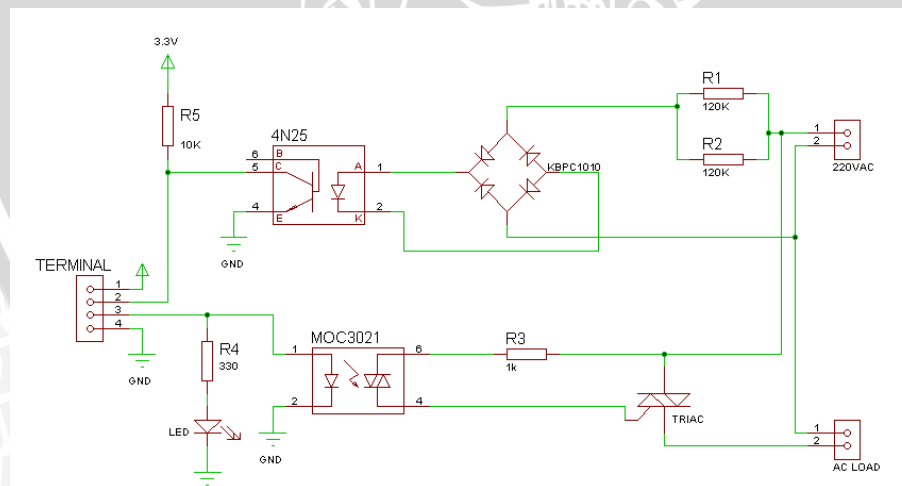
Sistem pemanas yang digunakan adalah dua (2) buah lampu pijar AC berdaya 60W untuk setiap lampunya. Pemasangan lampu diletakkan di dalam bagian atas mesin tetas, seperti pada Gambar 4.2.



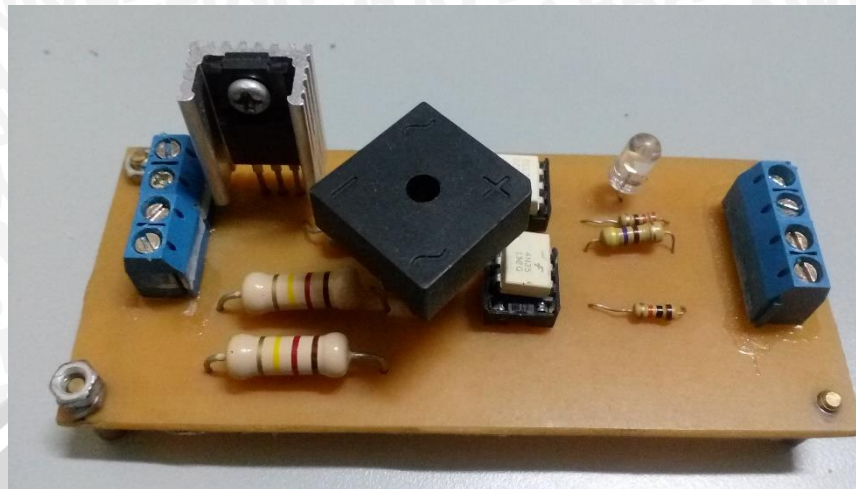
Gambar 4.2 Pemanas pada Mesin Tetes

4.5.2 Rangkaian Dimmer

Elemen pemanas yang digunakan adalah elemen pemanas dengan catu daya AC. Untuk mengendalikan besarnya tegangan yang melewati elemen pemanas digunakan rangkaian *dimmer* yang di dalamnya juga terdapat beberapa bagian, diantaranya adalah rangkaian pemecutan *gate* TRIAC dan rangkaian *Zero Cross Detector*. Gambar 4.3 menunjukkan skema rangkaian *dimmer* AC, sedangkan modul rangkaian yang akan dibuat ditunjukkan pada Gambar 4.4.



Gambar 4.3 Skema Rangkaian *Dimmer*



Gambar 4.4 Modul Rangkaian *Dimmer*

4.5.3 Permodelan Sensor SHT11

Jenis sensor yang digunakan untuk mengukur suhu dan kelembaban dalam penelitian ini adalah SHT11. Range pengukuran yang mampu dilakukan sensor SHT11 yaitu -40°C hingga $+123,8^{\circ}\text{C}$ atau dari -40°F hingga $+4,9^{\circ}\text{F}$ dan kelembaban relatif dari 0 %RH hingga 100 %RH. Sensor SHT11 membutuhkan catu daya +5V DC dengan konsumsi daya rendah 30 μW .

Nilai gain dari sensor dapat dihitung dengan Persamaan (4-1): (Seborg dkk,2004:214).

$$K_{\text{SHT}} = \frac{\text{range of instrument output}}{\text{range of instrument input}} \dots\dots\dots(4-1)$$

Sedangkan nilai *gain* dari sensor SHT11 suhu adalah:

$$K_{\text{SHT}} = \frac{5-0}{123,8-(-40)} = \frac{5}{163,8} = 0,04 \text{ V}/^{\circ}\text{C}$$

Sedangkan nilai *gain* dari sensor SHT11 kelembaban adalah:

$$K_{\text{SHT}} = \frac{5-0}{100-0} = \frac{5}{100} = 0,05 \text{ V}/\% \text{RH}$$

4.6 Perancangan Algoritma Kontrol Logika Fuzzy

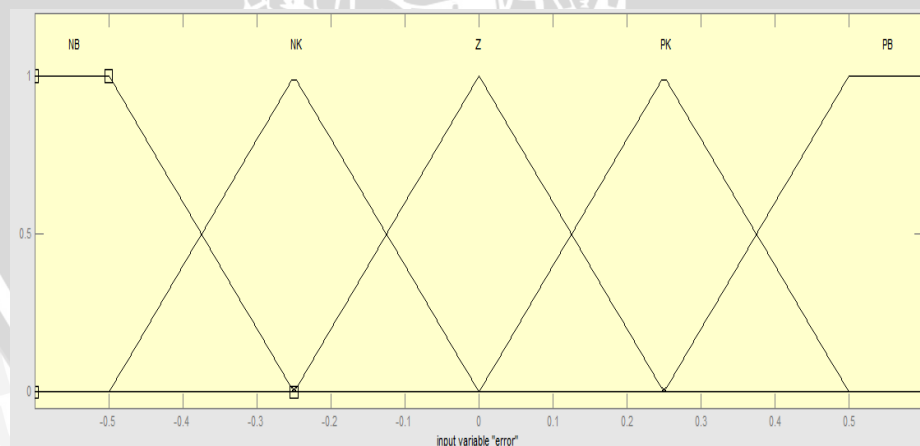
4.6.1 Perancangan Fungsi Keanggotaan

Fungsi keanggotaan dari *error*, *delta error* dan *output* suhu terdiri atas lima (5) label, yaitu Negatif Besar (NB), Negatif Kecil (NK), Zero (Z), Positif Kecil (PK), dan Positif Besar (PB) untuk *input* dan D (Dingin), KP (Kurang Panas), I (Ideal), KP (Kurang Panas), dan PS (Panas Sekali). Label masing-masing fungsi keanggotaan berjumlah lima dengan tujuan agar *rule* (aturan) yang dihasilkan semakin banyak, karena semakin banyak aturan maka semakin banyak kemungkinan kondisi yang tercipta dan semakin banyak pula aksi kontrol yang akan dipilih sehingga lebih sensitif terhadap perubahan *input* yang akan terjadi.

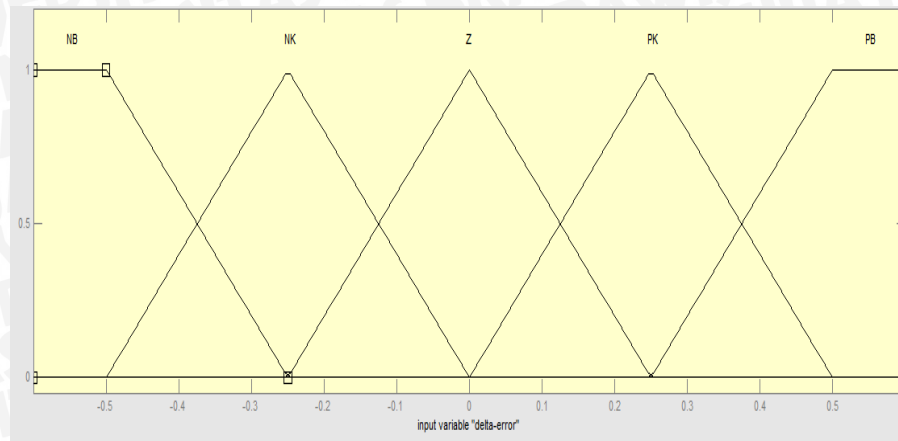
- **Perancangan Fungsi Keanggotaan Suhu**

Kontrol Logika Fuzzy yang dirancang memiliki dua buah masukan yaitu *error* dan *delta error*. Himpunan fuzzy untuk masukan *error* dan perubahan *error*, masing-masing dibagi menjadi 5 kelas untuk *error* dan 5 kelas untuk *delta error*. (lihat Gambar 4.5, 4.6, 4.7)

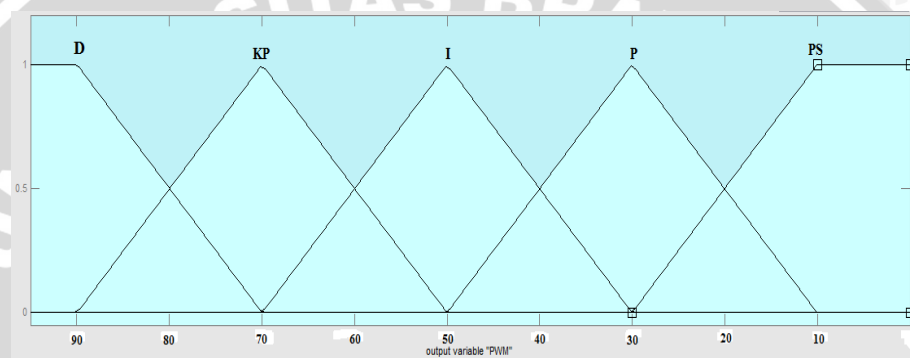
- *Error* adalah selisih antara *Setpoint* dan *Present Value* memiliki range -0,5 sampai 0,5.
- *Delta error* adalah selisih antara *error* dan *error(t-1)* memiliki range -0,5 sampai 0,5.
- Range suhu yaitu 37°C sampai 38 °C, *setpoint* suhu 37,5 °C.



Gambar 4.5 Perancangan *membership function error* suhu



Gambar 4.6 Perancangan *membership function* delta error suhu



Gambar 4.7 Perancangan *membership function* untuk PWM

Hubungan *error* dan *delta error* terhadap nilai PWM dapat dijabarkan sebagai berikut:

1. Saat nilai *error* relatif besar maka memperkecil nilai PWM *output*.
2. Saat nilai *error* dan *delta error* relatif sama maka nilai PWM diperbesar agar mengurangi *overshoot*.
3. Saat nilai *error* kecil maka nilai PWM diperbesar.

4.6.2 Perancangan *Rule Base*

Pada skripsi ini pembuatan *rule base* dan pengambilan keputusan terdiri dari dua masukan dan satu keluaran. Eksekusi aturan diproses menggunakan implikasi yang akan mengambil nilai paling minimal dari kedua masukan yaitu sinyal derajat keanggotaan *error* dan *delta error* dan juga menentukan konstanta parameter keluaran yang digunakan. *Input error* dan *delta error* masing-masing mempunyai 5 *membership functions*. Karena terdiri dari 2 masukan dan tiap masukan memiliki 5 *membership functions*, maka aturan yang mungkin terjadi

adalah 25 aturan. Tabel 4.1 menunjukkan *rule base fuzzy* yang digunakan dalam skripsi ini.

Tabel 4.1 Perancangan *Rule Base Fuzzy* Pengontrol Suhu

e \ de	NB	NK	Z	PK	PB
NB	D	D	KP	KP	I
NK	D	KP	KP	I	I
Z	KP	KP	I	I	P
PK	I	I	P	P	P
PB	P	P	PS	PS	PS

Keterangan :

1. D = Dingin
2. KP = Kurang Panas
3. I = Ideal
4. P = Panas
5. PS = Panas Sekali

4.7 Modul Arduino Mega 2560

Pada sistem pengendalian suhu ini digunakan Arduino Mega 2560 sebagai pengolah data dalam proses pengendalian elemen pemanas. Arduino Mega adalah *board* mikrokontroler berbasis ATmega1280. Memiliki 54 pin *input* dan *output* digital, di mana 14 pin tersebut dapat digunakan sebagai *output* PWM dan 16 pin *input* analog, 16 MHz osilator kristal, koneksi USB, *jack power*, ICSP *header*, dan tombol reset. Untuk dapat digunakan, mikrokontroler ini dihubungkan ke komputer dengan menggunakan kabel USB atau dengan adaptor AC-DC untuk menjalankannya.

Secara fisik, ukuran Arduino Mega hampir kurang lebih 2 kali lebih besar dari Arduino Uno, ini untuk mengakomodasi lebih banyaknya pin Digital dan Analog pada *board* Arduino Mega tersebut. Tampilan Arduino Mega dapat dilihat pada Gambar 4.8. Sedangkan penggunaan pin pada Arduino Mega disebutkan pada Tabel 4.1



Gambar 4.8 Modul Arduino Mega 2560

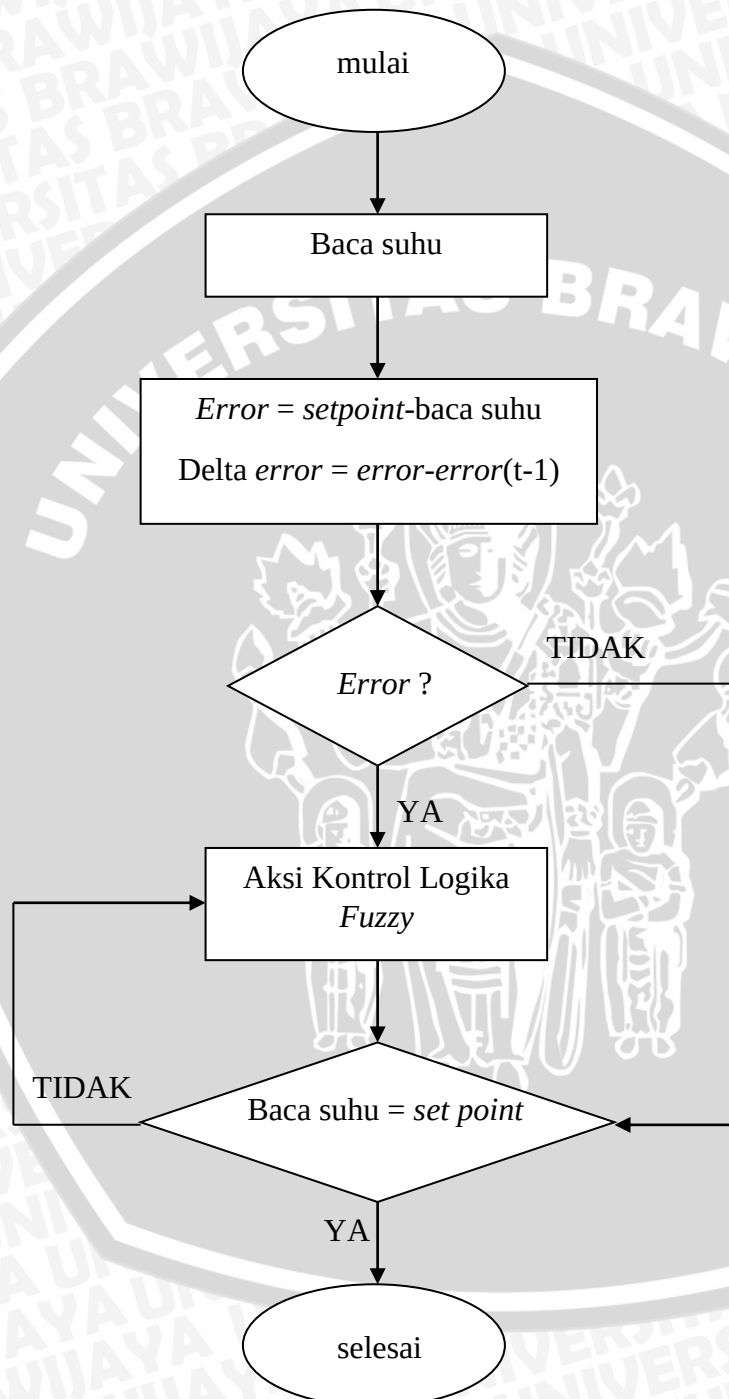
Tabel 4.2 Penggunaan Pin Pada Arduino Mega 2560

No.	Pin	Fungsi
1.	2	Jalur masukan <i>Zero Cross Interrupt</i>
2.	4	Jalur keluaran untuk rangkaian <i>dimmer</i>
3.	8	SHT11
4.	9	SHT11
5.	10	SHT11
6.	11	<i>Driver kipas</i>
7.	12	<i>Driver kipas</i>
8.	13	<i>Enable driver kipas</i>
9.	31	<i>Relay humidifier</i>
10.	22	LCD pin
11.	24	LCD pin
12.	26	LCD pin
13.	28	LCD pin
14.	30	LCD pin
15.	32	LCD pin
16.	34	LCD pin
17.	3,3V	<i>Vcc dimmer</i>
18.	5V	<i>Vcc SHT11, driver kipas, humidifier, LCD</i>
19.	GND	<i>Pin ground</i>

4.8 Flowchart Perancangan Perangkat Lunak

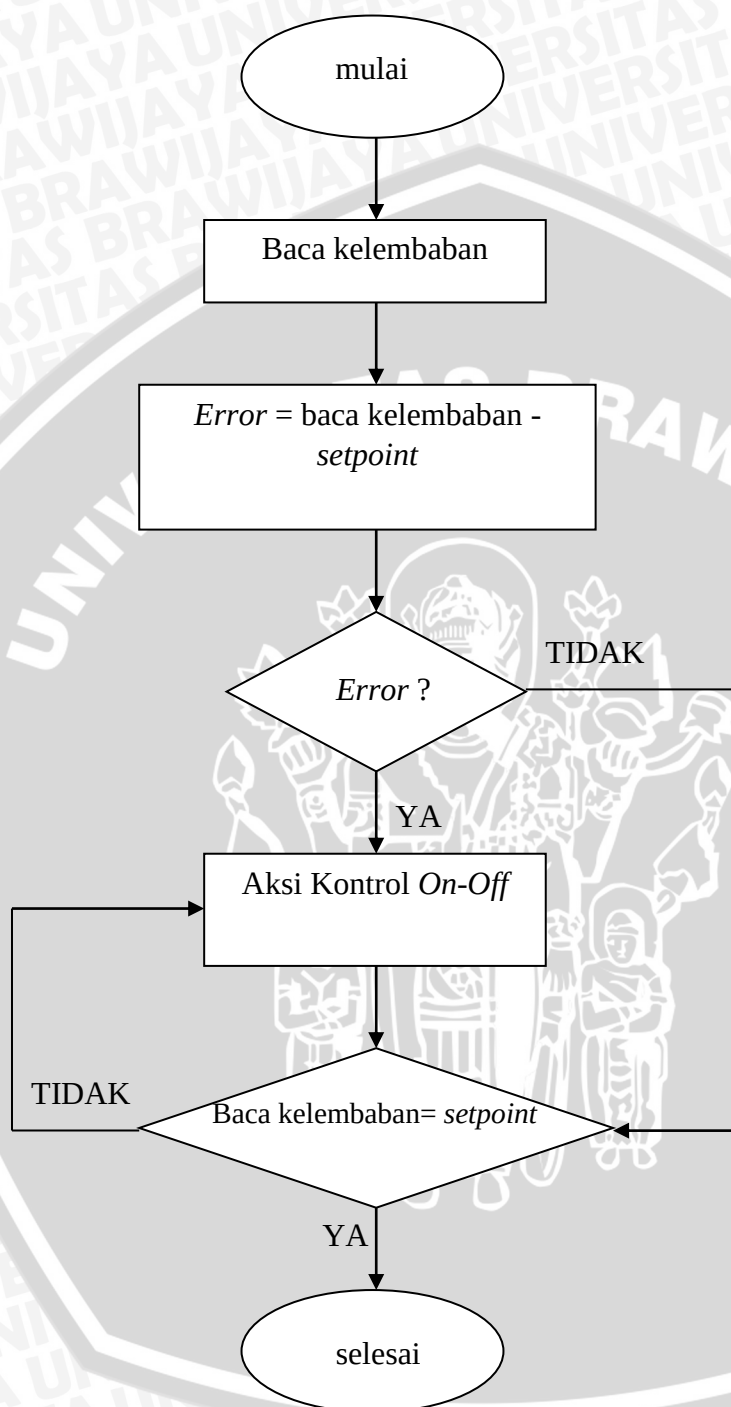
Flowchart perancangan perangkat lunak suhu dan kelembaban sebagai berikut.

4.8.1 Flowchart perancangan perangkat lunak suhu ditunjukkan pada Gambar 4.9.



Gambar 4.9 Flowchart Sistem Suhu

4.8.2 Flowchart perancangan perangkat lunak kelembaban ditunjukkan pada Gambar 4.10.



Gambar 4.10 Flowchart Sistem Kelembaban

BAB V

PENGUJIAN DAN ANALISIS SISTEM

Tujuan pengujian sistem ini 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. Pengujian pada sistem ini dibagi menjadi empat bagian, yaitu:

1. Pengujian sensor suhu dan kelembaban SHT11
2. Pengujian penyulutan TRIAC
3. Pengujian rangkaian *dimmer*
4. Pengujian sistem keseluruhan

5.1 Pengujian Sensor Suhu dan Kelembaban SHT11

Pengujian sensor suhu ditujukan untuk mengetahui kemampuan sensor SHT11 dalam membaca perubahan suhu.

5.1.1 Peralatan yang Digunakan

1. Sumber tegangan 5V DC.
2. Arduino Mega 2560.
3. Rangkaian Sensor Suhu dan Kelembaban SHT11.
4. 2 buah lampu pijar AC 60W sebagai pemanas (*heater*).
5. Komputer/Laptop

5.1.2 Prosedur Pengujian

1. Merangkai peralatan seperti blok diagram pada Gambar 5.1.
2. Mengunggah program pembacaan sensor suhu pada Arduino.
3. Mengamati dan mencatat nilai pada tampilan serial monitor.



Gambar 5.1 Blok Pengujian Sensor SHT11 untuk Suhu

5.1.3 Hasil Pengujian

Hasil pengujian pembacaan suhu yang ditampilkan pada serial monitor dan dibandingkan pada pembacaan suhu dari termometer ditunjukkan pada Tabel 5.1.

Tabel 5.1 Hasil Pengujian Sensor Suhu SHT11

No	Suhu Termometer (°C)	Suhu Sensor SHT 11(°C)
1	34,5	34,7
2	35	35,06
3	35,5	35,6
4	36	36,07
5	36,5	36,59
6	37	37,05
7	37,5	37,55

Perhitungan kesalahan pembacaan sensor:

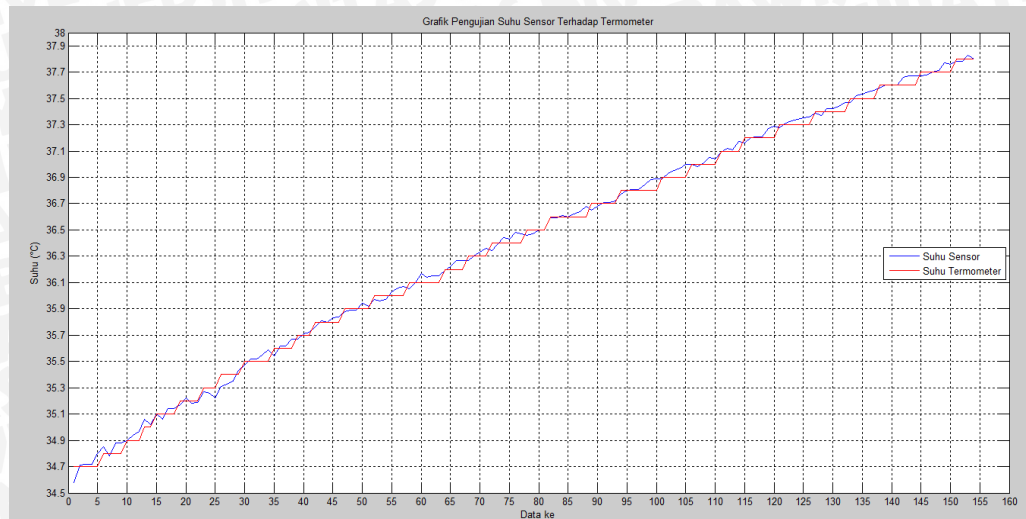
$$\% \text{ kesalahan} = \frac{(\sum \text{Hasil Pengukuran} - \sum \text{Referensi})}{\sum \text{Referensi}} \times 100\%$$

$$\% \text{ kesalahan} = \frac{(5605,67 - 5603,3)}{5603,3} \times 100\%$$

$$= 0,0422\%$$

Dapat disimpulkan bahwa sensor SHT11 untuk pembacaan suhu dapat bekerja dengan baik.

Grafik hubungan antara suhu sensor SHT11 dengan termometer digital dapat dilihat pada Gambar 5.2.



Gambar 5.2 Perbandingan Suhu Sensor SHT11 terhadap Termometer Digital

5.2 Pengujian Sensor Kelembaban SHT11

Pengujian sensor SHT11 ditujukan untuk mengetahui apakah sensor dapat membaca kelembaban dengan baik.

5.2.1. Peralatan yang Digunakan

1. Sumber tegangan 5V DC.
2. Arduino Mega 2560.
3. *Humidifier* USB 5V DC
4. *Relay Module*.
5. Komputer.

5.2.2. Prosedur Pengujian

1. Merangkai peralatan seperti blok diagram pada Gambar 5.3.
2. Mengunggah program pembacaan sensor kelembaban pada arduino.
3. Mengamati dan mencatat nilai pada tampilan serial monitor.



Gambar 5.3 Blok Pengujian Sensor SHT11 untuk Kelembaban

5.2.3 Hasil Pengujian

Hasil pengujian pembacaan kelembaban yang ditampilkan pada serial monitor dan dibandingkan dengan pembacaan dari hygrometer ditunjukkan pada Tabel 5.2.

Tabel 5.2 Hasil Pengujian Sensor Kelembaban SHT11

No	Kelembaban Hygrometer (%RH)	Kelembaban Sensor SHT11 (%RH)
1	64	62,8
2	66	65,04
3	68	66,67
4	70	70,44
5	72	72,5
6	74	73,89
7	76	76,43
8	78	78,61
9	80	80,58
10	82	82,11
11	84	83,80

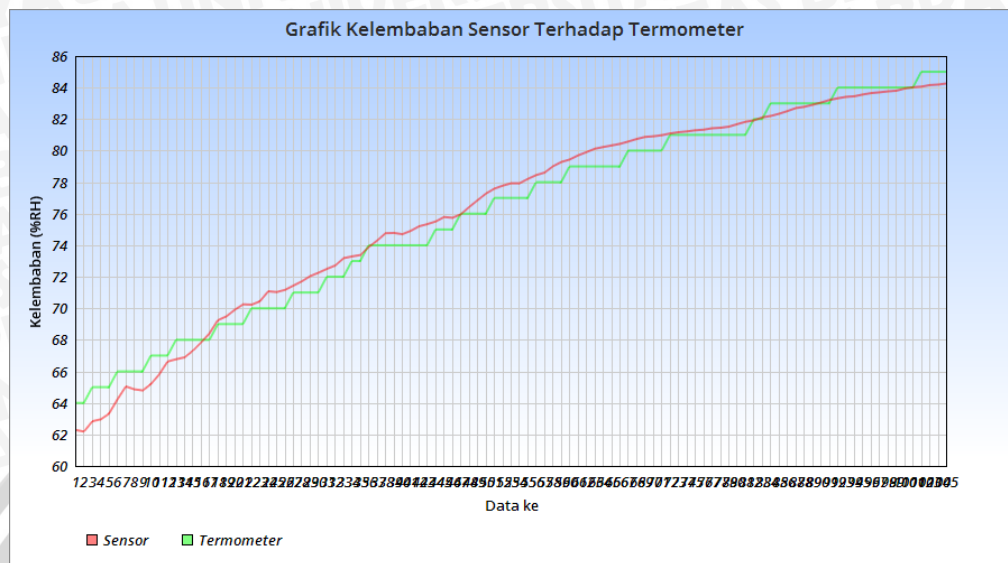
Perhitungan kesalahan pembacaan sensor:

$$\% \text{ kesalahan} = \frac{(\sum \text{Hasil Pengukuran} - \sum \text{Referensi})}{\sum \text{Referensi}} \times 100\%$$

$$\begin{aligned} \% \text{ kesalahan} &= \frac{(8090,53 - 8074)}{8074} \times 100\% \\ &= 0.2047\% \end{aligned}$$

Dapat disimpulkan bahwa sensor SHT11 untuk pembacaan kelembaban dapat bekerja dengan baik.

Grafik hubungan antara suhu sensor SHT11 dengan hygrometer digital dapat dilihat pada Gambar 5.4.



Gambar 5.4 Grafik Pengujian Sensor Kelembaban SHT11

5.3 Pengujian Penyulutan TRIAC

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

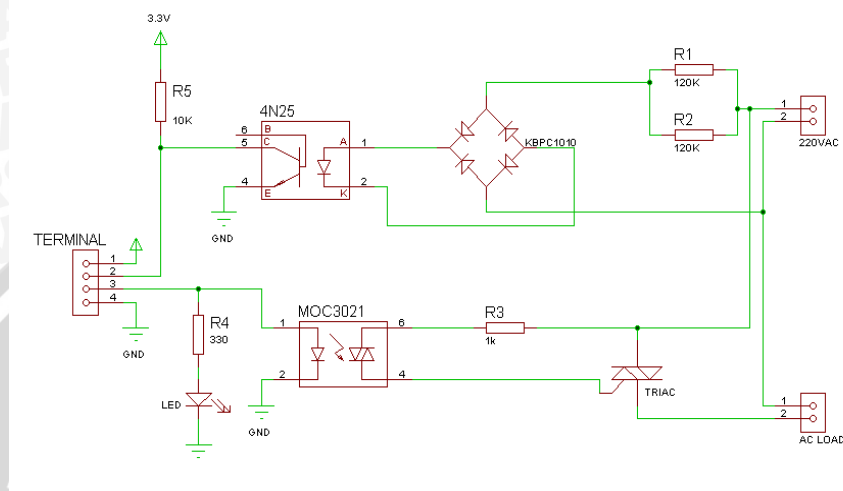
5.3.1 Peralatan yang Digunakan

1. Sumber Tegangan 220 VAC.
2. Arduino Mega 2560.
3. Rangkaian *Zero Cross Detector*.
4. Rangkaian TRIAC dan MOC3021.
5. Oscilloscope.
6. Multimeter.
7. Komputer.

5.3.2 Prosedur Pengujian

1. Membuat rangkaian seperti Gambar 5.5.
2. Menghubungkan sumber tegangan 5V ke Arduino dan ke optocoupler 4N25 (terminal 1).
3. Menghubungkan rangkaian ke sumber tegangan 220 VAC.
4. Menghubungkan keluaran optocoupler 4N25 ke pin *interrupt* Arduino (terminal 2).

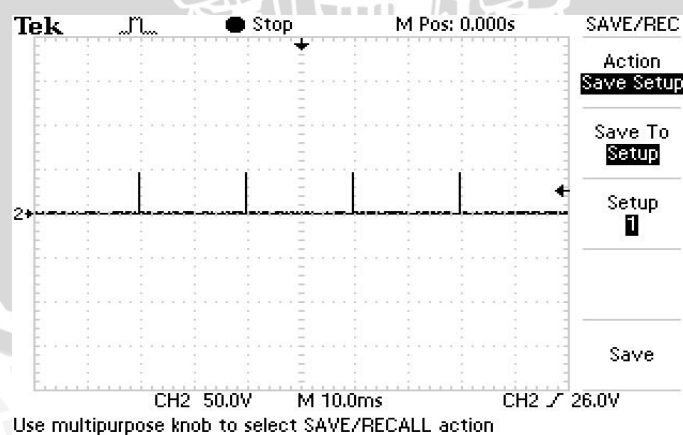
5. Menghubungkan salah satu pin digital I/O dari arduino ke MOC3021 (terminal 3).
6. Menghubungkan *probe* 1 oscilloscope ke keluaran Arduino (pin digital I/O) dan *probe* 2 ke tegangan keluaran TRIAC (AC Load).



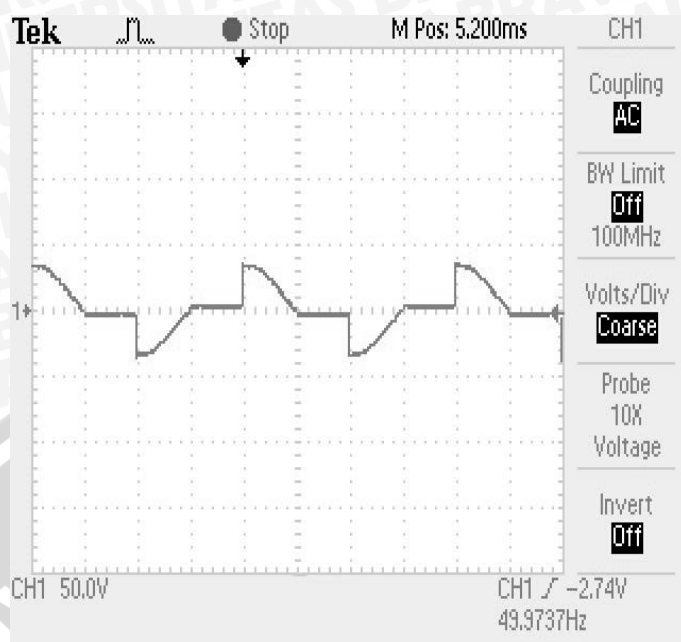
Gambar 5.5 Rangkaian Pengujian Sudut Penyalaan TRIAC

5.3.3 Hasil Pengujian

Dari pengujian penyalaan TRIAC didapatkan hasil pengujian berupa bentuk gelombang keluaran TRIAC sesuai dengan sudut penyalannya. Contoh hasil gelombang keluaran arduino dan triac sudut penyalaan 45° atau PWM 35 ditunjukkan pada Gambar 5.6 dan Gambar 5.7.



Gambar 5.6 Bentuk Gelombang Keluaran Arduino



Gambar 5.7 Bentuk gelombang Keluaran TRIAC dengan PWM 32

Tabel 5.3 Hasil Pengujian Sudut Penyalaan TRIAC

PWM	Sudut Penyalaan (°)	Perhitungan waktu tunda (ms)	Pengujian waktu tunda (ms)
16	22,5	1,25	1,1
32	45	2,5	2,4
48	67,5	3,75	3,5
64	90	5	4,8
80	112,5	6,25	6
96	135	7,5	7,2
112	157,5	8,75	8,4
128	180	10	9,8

Berdasarkan data hasil pengujian sudut penyalaan TRIAC yang didapatkan pada Tabel 5.3 terjadi penyimpangan waktu tunda dengan data hasil perhitungan.

- Perhitungan pengujian data ke-1:

$$\begin{aligned} \text{Waktu tunda (ms)} &= \frac{\text{Sudut Penyalaan}}{180^\circ} \times 10 \\ &= \frac{22,5^\circ}{180^\circ} \times 10 \end{aligned}$$

$$= 1,25 \text{ ms}$$

Nilai kesalahan yang didapat dari selisih absolut perhitungan dan pengujian sebesar:

$$\begin{aligned} \text{Kesalahan (ms)} &= |\text{Perhitungan} - \text{Pengujian}| \\ &= |1,25 - 1,1| = 0,15 \end{aligned}$$

$$\begin{aligned} \text{Kesalahan (\%)} &= \frac{\text{Kesalahan}}{\text{Data Perhitungan}} \times 100\% \\ &= \frac{0,15}{1,25} \times 100\% \\ &= 12\% \end{aligned}$$

- Perhitungan pengujian data ke-2:

$$\begin{aligned} \text{Waktu tunda (ms)} &= \frac{\text{Sudut Penyalaan}}{180^\circ} \times 10 \\ &= \frac{45^\circ}{180^\circ} \times 10 = 2,5 \text{ ms} \end{aligned}$$

Nilai kesalahan yang didapat dari selisih absolut perhitungan dan pengujian sebesar:

$$\begin{aligned} \text{Kesalahan (ms)} &= |\text{Perhitungan} - \text{Pengujian}| \\ &= |2,5 - 2,4| = 0,1 \end{aligned}$$

$$\begin{aligned} \text{Kesalahan (\%)} &= \frac{\text{Kesalahan}}{\text{Data Perhitungan}} \times 100\% \\ &= \frac{0,1}{2,5} \times 100\% \\ &= 4\% \end{aligned}$$

- Perhitungan pengujian data ke-3:

$$\begin{aligned} \text{Waktu tunda (ms)} &= \frac{\text{Sudut Penyalaan}}{180^\circ} \times 10 \\ &= \frac{67,5^\circ}{180^\circ} \times 10 \\ &= 3,75 \text{ ms} \end{aligned}$$

Nilai kesalahan yang didapat dari selisih absolut perhitungan dan pengujian sebesar:

$$\begin{aligned} \text{Kesalahan (ms)} &= |\text{Perhitungan} - \text{Pengujian}| \\ &= |3,75 - 3,5| = 0,25 \end{aligned}$$

$$\begin{aligned} \text{Kesalahan (\%)} &= \frac{\text{Kesalahan}}{\text{Data Perhitungan}} \times 100\% \\ &= \frac{0,25}{3,75} \times 100\% \end{aligned}$$

$$= 6,67\%$$

- Perhitungan pengujian data ke-4:

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

Nilai kesalahan yang didapat dari selisih absolut perhitungan dan pengujian sebesar:

$$\begin{aligned}\text{Kesalahan (ms)} &= |\text{Perhitungan} - \text{Pengujian}| \\ &= |5 - 4,8| = 0,2\end{aligned}$$

$$\begin{aligned}\text{Kesalahan (\%)} &= \frac{\text{Kesalahan}}{\text{Data Perhitungan}} \times 100\% \\ &= \frac{0,2}{5} \times 100\% \\ &= 4\%\end{aligned}$$

- Perhitungan pengujian data ke-5:

$$\begin{aligned}\text{Waktu tunda (ms)} &= \frac{\text{Sudut Penyalaan}}{180^\circ} \times 10 \\ &= \frac{112,5^\circ}{180^\circ} \times 10 \\ &= 6,25 \text{ ms}\end{aligned}$$

Nilai kesalahan yang didapat dari selisih absolut perhitungan dan pengujian sebesar:

$$\begin{aligned}\text{Kesalahan (ms)} &= |\text{Perhitungan} - \text{Pengujian}| \\ &= |6,25 - 6| = 0,25\end{aligned}$$

$$\begin{aligned}\text{Kesalahan (\%)} &= \frac{\text{Kesalahan}}{\text{Data Perhitungan}} \times 100\% \\ &= \frac{0,25}{6,25} \times 100\% \\ &= 4,00\%\end{aligned}$$

- Perhitungan pengujian data ke-6:

$$\begin{aligned}\text{Waktu tunda (ms)} &= \frac{\text{Sudut Penyalaan}}{180^\circ} \times 10 \\ &= \frac{135^\circ}{180^\circ} \times 10 \\ &= 7,5 \text{ ms}\end{aligned}$$

Nilai kesalahan yang didapat dari selisih absolut perhitungan dan pengujian sebesar:

$$\begin{aligned}\text{Kesalahan (ms)} &= |\text{Perhitungan} - \text{Pengujian}| \\ &= |7,5 - 7,2| = 0,3\end{aligned}$$

$$\begin{aligned}\text{Kesalahan (\%)} &= \frac{\text{Kesalahan}}{\text{Data Perhitungan}} \times 100\% \\ &= \frac{0,3}{7,5} \times 100\% \\ &= 4,00\%\end{aligned}$$

- Perhitungan pengujian data ke-7:

$$\begin{aligned}\text{Waktu tunda (ms)} &= \frac{\text{Sudut Penyalaan}}{180^\circ} \times 10 \\ &= \frac{157,5^\circ}{180^\circ} \times 10 \\ &= 8,75 \text{ ms}\end{aligned}$$

Nilai kesalahan yang didapat dari selisih absolut perhitungan dan pengujian sebesar:

$$\begin{aligned}\text{Kesalahan (ms)} &= |\text{Perhitungan} - \text{Pengujian}| \\ &= |8,75 - 8,4| = 0,35\end{aligned}$$

$$\begin{aligned}\text{Kesalahan (\%)} &= \frac{\text{Kesalahan}}{\text{Data Perhitungan}} \times 100\% \\ &= \frac{0,35}{8,75} \times 100\% \\ &= 4,00\%\end{aligned}$$

- Perhitungan pengujian data ke-8:

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

Nilai kesalahan yang didapat dari selisih absolut perhitungan dan pengujian sebesar:

$$\begin{aligned}\text{Kesalahan (ms)} &= |\text{Perhitungan} - \text{Pengujian}| \\ &= |10 - 9,8| = 0,4\end{aligned}$$

$$\begin{aligned}\text{Kesalahan (\%)} &= \frac{\text{Kesalahan}}{\text{Data Perhitungan}} \times 100\% \\ &= \frac{0,2}{10} \times 100\% \\ &= 2,00\%\end{aligned}$$

$$= 2\%$$

Hasil perhitungan kesalahan pengujian secara keseluruhan dapat dilihat pada Tabel

5.4.

Tabel 5.4 Kesalahan Pengujian PWM Penyalan TRIAC

PWM	Perhitungan waktu tunda (ms)	Pengujian waktu tunda (ms)	Kesalahan (ms)	Kesalahan (%)
16	1,25	1,1	0,15	12,00
32	2,5	2,4	0,1	4,00
48	3,75	3,5	0,25	6,67
64	5	4,8	0,2	4,00
80	6,25	6	0,25	4,00
96	7,5	7,2	0,3	4,00
112	8,75	8,4	0,35	4,00
128	10	9,8	0,2	2,00
Rata Rata Kesalahan (%)				5,084

Dari hasil perhitungan kesalahan pengujian secara keseluruhan didapatkan nilai rata-rata kesalahan keseluruhan sebesar 5,084%. Kesalahan ini dikarenakan adanya proses perhitungan dan pembulatan angka oleh mikrokontroler dan sumber AC yang tidak ideal baik dari frekuensi dan bentuk gelombang tegangan, sehingga menyebabkan adanya selisih antara perhitungan dan pengukuran. Berdasarkan analisis di atas bisa disimpulkan bahwa mikrokontroler mampu menghasilkan sudut penyalan yang sesuai dengan yang diberikan.

5.4 Pengujian Rangkaian *Dimmer*

Pengujian ini dilakukan untuk mengetahui apakah rangkaian *dimmer* dapat digunakan sebagai *driver* untuk lampu pijar.

5.4.1 Peralatan yang Digunakan

1. Sumber Tegangan AC.
2. Arduino Mega 2560.
3. Rangkaian *Dimmer*.
4. Multimeter.

5. Komputer.

5.4.2 Prosedur Pengujian

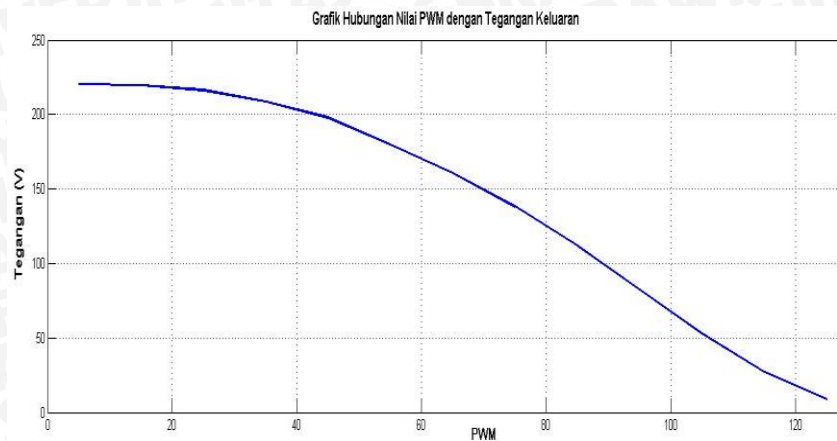
1. Menghubungkan rangkaian *dimmer* dengan arduino.
2. Menghubungkan rangkaian *dimmer* dengan sumber AC.
3. Menghubungkan Arduino dengan komputer.
4. Memberikan beban lampu pada rangkaian *dimmer*.
5. Memberikan nilai PWM pada rangkaian *dimmer* melalui Arduino.
6. Mengukur tegangan keluaran rangkaian *dimmer* kepada beban.

5.4.3 Hasil Pengujian

Dari hasil pengujian rangkaian *dimmer* didapatkan hasil berupa nilai tegangan yang berbeda-beda sesuai dengan nilai PWM yang diberikan. Hasil pengujian dapat dilihat pada Tabel 5.5

Tabel 5.5 Perbandingan Nilai PWM Terhadap Tegangan

No.	PWM	Tegangan (V)
1	125	8,92
2	115	27,43
3	105	52,8
4	95	82,5
5	85	111,8
6	75	138,4
7	65	161,1
8	55	180
9	45	198,2
10	35	208,7
11	25	216,5
12	5	220,7



Gambar 5.8 Grafik Hubungan Nilai PWM dengan Tegangan

Dari Tabel 5.5 dapat disimpulkan bahwa rangkaian *dimmer* dapat berfungsi dengan baik dan dapat digunakan sebagai *driver* aktuator (lampu pijar), ditandai dengan adanya perubahan tegangan akibat perubahan nilai PWM yang diberikan.

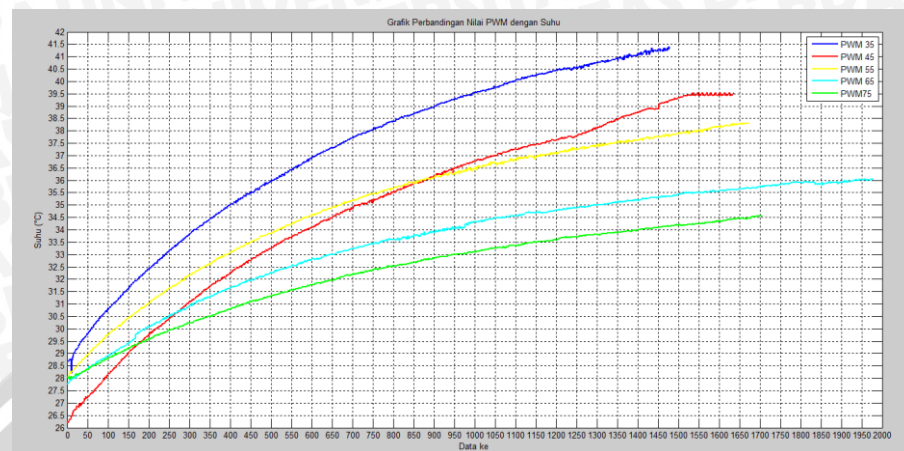
Dari hasil pengujian rangkaian *dimmer* didapatkan juga hasil nilai berupa suhu *steadystate* yang berbeda-beda sesuai dengan nilai PWM yang diberikan. Hasil pengujian dapat dilihat pada Tabel 5.6

Tabel 5.6 Perbandingan Nilai PWM Terhadap *steadystate* Suhu

No.	PWM	Suhu <i>Steadystate</i> (°C)
1	10	41,3
2	30	39,54
3	50	38,31
4	70	36,02
5	90	34,53

Dari Tabel 5.8 dapat disimpulkan bahwa rangkaian *dimmer* dapat digunakan sebagai *driver* aktuator (lampu pijar), ditandai dengan adanya perubahan suhu *steadystate* akibat perubahan nilai PWM yang diberikan.

Dari hasil pengujian rangkaian *dimmer* didapatkan hasil nilai karakteristik grafik respon plant mesin tetas. Hasil pengujian dapat dilihat pada Gambar 5.9.



Gambar 5.9 Karakteristik Plant

Dari Gambar 5.9 dapat disimpulkan bahwa grafik karakteristik respon pada rangkaian *dimmer* dapat digunakan sebagai *driver* aktuator pemanas, ditandai dengan adanya perubahan suhu *steadystate* akibat perubahan nilai PWM yang diberikan.

5.5 Pengujian Keseluruhan Sistem

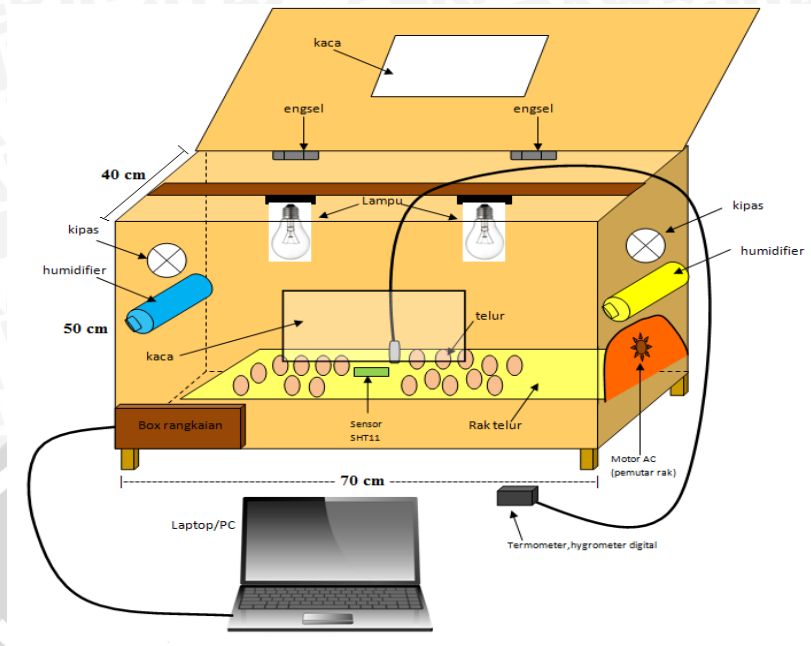
Pengujian ini dilakukan untuk mengetahui seberapa besar nilai kontroler yang dibutuhkan agar sistem bekerja sesuai dengan setpoint 37,5°C dan kelembaban 55 – 65 %RH.

5.5.1 Peralatan yang Digunakan

1. *Plant* mesin tetas.
2. Sensor suhu dan kelembaban SHT11.
3. Rangkaian *Dimmer*.
4. *Humidifier* 5V
5. Kipas DC 12V
6. Lampu Pijar AC 60W
7. Arduino Mega 2560.
8. Sumber tegangan AC.
9. Komputer.

5.5.2 Prosedur Pengujian

1. Menghubungkan rangkaian seperti Gambar 5.10

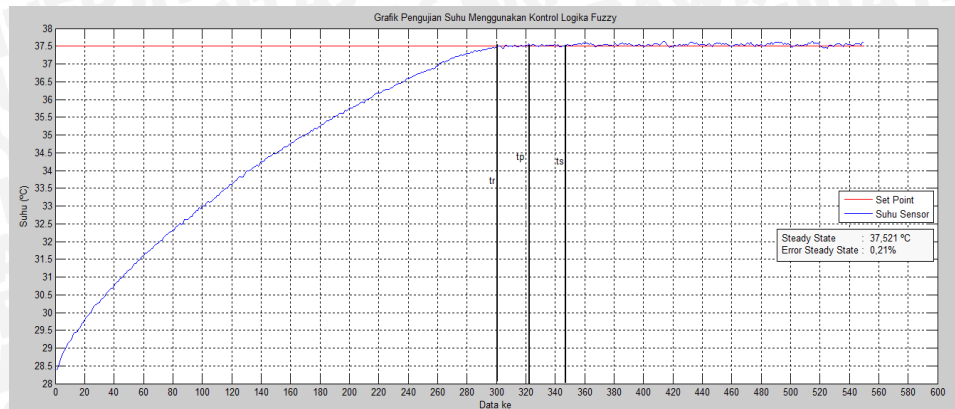


Gambar 5.10 Sistem Keseluruhan

2. Menghubungkan *power supply* sebagai sumber caru untuk Arduino Mega 2560 dan komponen lainnya yang membutuhkan catu daya.
3. Mengunduh program dengan Kontrol Logika *Fuzzy* dan *On-Off* sesuai dengan parameter yang telah didapat melalui *software* Arduino 1.6.5.
4. Setelah semua komponen dan rangkaian dipastikan terpasang dengan benar maka program dijalankan.
5. Mengamati dan menganalisa hasil kinerja sistem.
6. Pada pengujian pertama kinerja sistem tidak diberi gangguan, pada pengujian kedua kinerja sistem diberi gangguan dengan cara membuka penutup mesin tetas selama 5 menit sehingga dapat menurunkan suhu ruangan mesin tetas.
7. Membuat grafik dari data yang didapat dari hasil pengujian sistem.

5.5.3 Hasil Pengujian

Pengujian untuk pengendalian suhu dengan menggunakan Kontrol Logika *Fuzzy* didapatkan hasil respon yang ditunjukkan pada Gambar 5.11.



Gambar 5.11 Grafik Respon Sistem Suhu dengan Kontrol Logika Fuzzy

Pada Gambar 5.11 saat keadaan *steady state* dengan nilai suhu tengah *steady state* ($T_{\text{tengah steady state}}$) 37,521°C dengan suhu mula-mula ($T_{\text{mula-mula}}$) adalah 28,39°C, serta suhu *setpoint* (T_{setpoint}) adalah 37,5°C, sehingga *error steady state* nya adalah:

$$e_{ss}(\%) = \left| \frac{(T_{\text{tengah steady state}}) - (T_{\text{setpoint}})}{(T_{\text{setpoint}}) - (T_{\text{mula-mula}})} \right| \times 100\%$$

$$= \left| \frac{37,521 - 37,5}{37,5 - 28,39} \right| \times 100\%$$

$$= 0,21\%$$

Maka diperoleh data pada Tabel 4.

Tabel 5.7 Data Respon Suhu Menggunakan Kontrol Logika Fuzzy

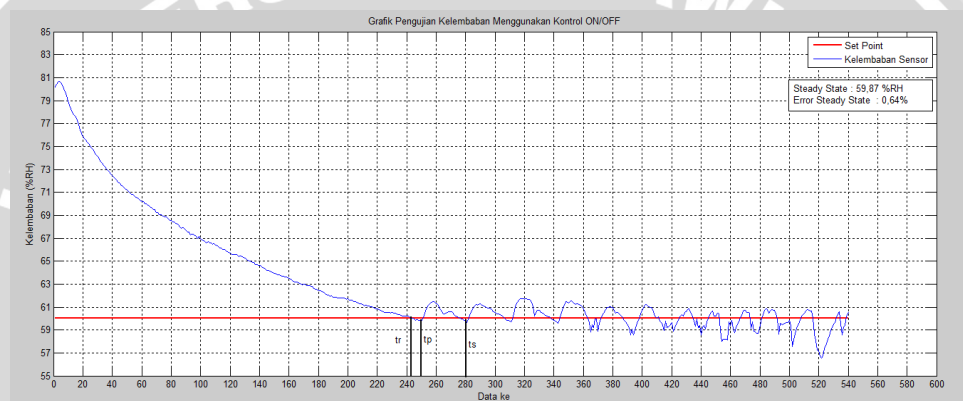
Waktu naik (t_r)	600 detik
Waktu Puncak (t_p)	644 detik
Waktu Keadaan Mantap (t_s)	690 detik
Error Steady State (E_{ss})	0,21%

Keterangan:

- e_{ss} : Error Steady State
- $T_{\text{tengah steady state}}$: Suhu rata-rata dari data ke-345 sampai dengan data ke-~.
- $T_{\text{mula-mula}}$: Suhu saat data pertama atau waktu ke-0 detik.
- T_{setpoint} : Suhu yang diinginkan, yaitu 37,5°C.

- Waktu Naik (t_r) adalah waktu yang diperlukan respons untuk naik dari 0-100% dari nilai akhirnya.
- Waktu Puncak (t_p) adalah waktu yang diperlukan respon untuk mencapai nilai puncak pertama respon.
- Waktu Keadaan Mantap (t_s) adalah waktu yang dibutuhkan respons untuk mencapai keadaan *steady state*.
- Data keluar setiap 2 detik.

Pada pengujian untuk pengendalian kelembaban dengan menggunakan kontroler ON-OFF dan *setpoint* kelembaban sebesar 60 %RH didapatkan hasil respon sistem yang ditunjukkan pada Gambar 5.12.



Gambar 5.12 Grafik Respon Sistem Kelembaban 60%RH dengan kontroler *On-Off*.

$$\begin{aligned}
 e_{ss} (\%) &= \left| \frac{(H \text{ tengah steady state}) - (H \text{ setpoint})}{(H \text{ setpoint}) - (H \text{ mula-mula})} \right| \times 100\% \\
 &= \left| \frac{59,87 - 60}{60 - 80,14} \right| \times 100\% \\
 &= 0,64\%
 \end{aligned}$$

Maka diperoleh data pada tabel 5.8.

Tabel 5.8 Data Respon Kelembaban 60%RH Menggunakan Kontroler *On-Off*

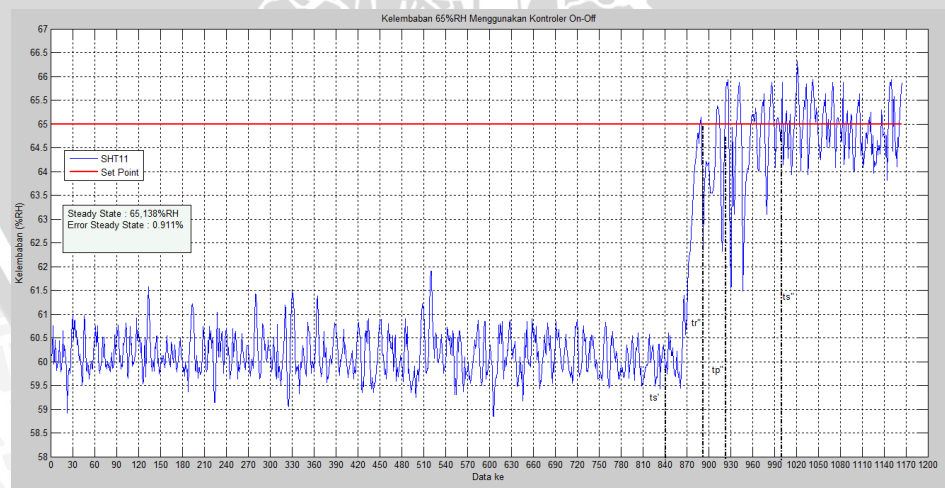
Waktu Naik (t_r)	484 detik
Waktu Puncak (t_p)	500 detik
Waktu Keadaan Mantap (t_s)	560 detik
<i>Error Steady State</i>	0,64%

Keterangan:

- e_{ss} : *Error Steady State*
- $H_{\text{tengah steady state}}$: Kelembaban rata-rata dari data ke-280 sampai dengan data ke-~.
- $H_{\text{mula-mula}}$: Kelembaban saat data pertama atau waktu ke-0 detik.
- H_{setpoint} : Kelembaban yang diinginkan, yaitu 60%RH.
- Waktu Naik (t_r) adalah waktu yang diperlukan respons untuk naik dari 0-100% dari nilai akhirnya.
- Waktu Puncak (t_p) adalah waktu yang diperlukan respon untuk mencapai nilai puncak pertama respon.
- Waktu Keadaan Mantap (t_s) adalah waktu yang dibutuhkan respons untuk mencapai keadaan *steady state*.
- Data keluar setiap 2 detik.

Tanggapan atau respon kelembaban menggunakan kontroler *On-Off* dengan *setpoint* 60%RH berada pada *setpoint* yang diinginkan.

Pada pengujian untuk pengendalian kelembaban dengan menggunakan kontroler *On-Off* dan *setpoint* kelembaban sebesar 65 %RH didapatkan hasil respon sistem yang ditunjukkan pada Gambar 5.13.



Gambar 5.13 Grafik Respon Sistem Kelembaban 65%RH dengan kontroler *On-Off*.

$$e_{ss} (\%) = \left| \frac{(H \text{ tengah steady state}) - (H \text{ setpoint})}{(H \text{ setpoint}) - (H \text{ mula-mula})} \right| \times 100\%$$

$$= \left| \frac{65,138 - 65}{65 - 80,14} \right| \times 100\%$$

$$= 0,911\%$$

Maka diperoleh data pada Tabel 5.9.

Tabel 5.9 Data Respon Kelembaban 65%RH Menggunakan Kontroler *On-Off*

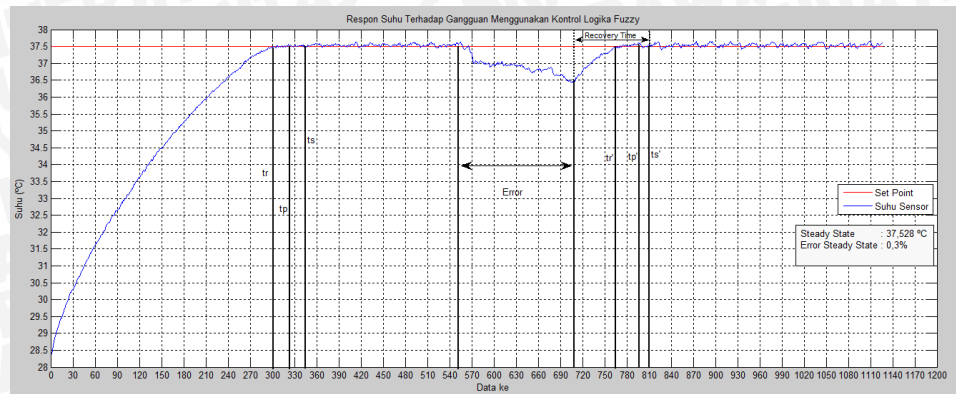
Waktu Naik (t_r)	100 detik
Waktu Puncak (t_p)	160detik
Waktu Keadaan Mantap (t_s)	320 detik
<i>Error Steady State</i>	0,911%

Keterangan:

- e_{ss} : *Error Steady State*
- $H \text{ tengah steady state}$: Kelembaban rata-rata dari data ke-1000 sampai dengan data ke-~.
- $H \text{ mula-mula}$: Kelembaban saat data pertama ke-840 detik
- $H \text{ setpoint}$: Kelembaban yang diinginkan, yaitu 65%RH
- Waktu Naik (t_r) adalah waktu yang diperlukan respons untuk naik dari 0-100% dari nilai akhirnya.
- Waktu Puncak (t_p) adalah waktu yang diperlukan respon untuk mencapai nilai puncak pertama respon.
- Waktu Keadaan Mantap (t_s) adalah waktu yang dibutuhkan respons untuk mencapai keadaan *steady state*.
- Data keluar setiap 2 detik.

Tanggapan atau respon kelembaban menggunakan kontroler *On-Off* dengan *setpoint* 65%RH berada pada *setpoint* yang diinginkan.

Pengujian sistem suhu menggunakan Kontrol Logika Fuzzy dan *setpoint* suhu sebesar 37,5°C dengan diberi gangguan berupa penutup mesin tetas dibuka selama 5 menit. Didapatkan hasil respon sistem sistem yang ditunjukkan pada Gambar 5.14.



Gambar 5.14 Grafik Respon Sistem Suhu Terhadap Gangguan Menggunakan Kontrol Logika Fuzzy

$$e_{ss} (\%) = \left| \frac{(T \text{ tengah steady state}) - (T \text{ setpoint})}{(T \text{ setpoint}) - (T \text{ mula-mula})} \right| \times 100\%$$

$$= \left| \frac{37,528 - 37,5}{37,5 - 28,39} \right| \times 100\%$$

$$= 0,3\%$$

Maka diperoleh data pada Tabel 5.10

Tabel 5.10 Data Respon Suhu Terhadap Gangguan Menggunakan Kontrol Logika Fuzzy

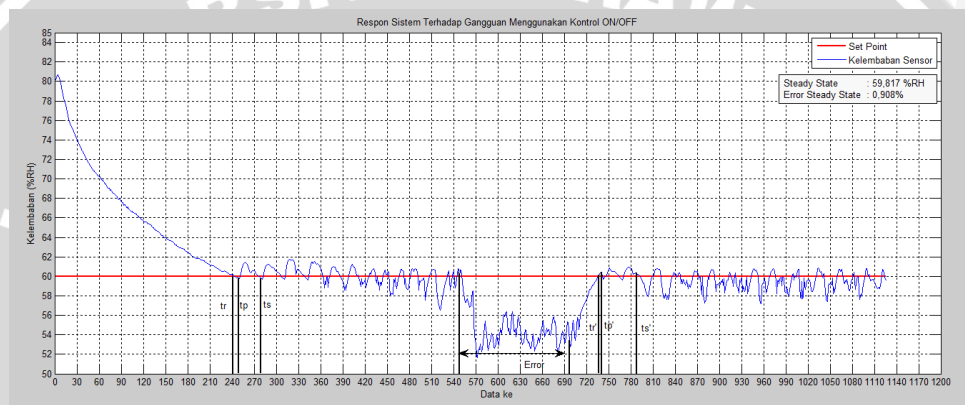
Waktu Naik (tr)	116 detik
Waktu Puncak (tp)	180 detik
Waktu Keadaan Mantap (ts)	206 detik
Error Steady State	0,3%

Keterangan:

- e_{ss} : Error Steady State
- $T \text{ tengah steady state}$: Suhu rata-rata dari data ke-810 sampai dengan data ke-~.
- $T \text{ mula-mula}$: Suhu saat data pertama atau waktu ke-0 detik
- $T \text{ setpoint}$: Suhu yang diinginkan, yaitu 37,5°C
- Waktu Naik (t_r) adalah waktu yang diperlukan respons untuk naik dari 0-100% dari nilai akhirnya.

- Waktu Puncak (t_p) adalah waktu yang diperlukan respon untuk mencapai nilai puncak pertama respon.
- Waktu Keadaan Mantap (t_s) adalah waktu yang dibutuhkan respons untuk mencapai keadaan *steady state*.
- Data keluar setiap 2 detik.

Pengujian sistem kelembaban menggunakan kontroler *On-Off* dan *setpoint* kelembaban sebesar 60%RH dengan diberi gangguan berupa penutup mesin tetas dibuka selama 5 menit. Didapatkan hasil respon sistem sistem yang ditunjukkan pada Gambar 5.15.



Gambar 5.15 Respon Sistem Kelembaban 60%RH Terhadap Gangguan Menggunakan Kontroler *On-Off*.

$$e_{ss} (\%) = \left| \frac{(H \text{ tengah steady state}) - (H \text{ setpoint})}{(H \text{ setpoint}) - (H \text{ mula-mula})} \right| \times 100\%$$

$$= \left| \frac{59,817 - 60}{60 - 80,14} \right| \times 100\%$$

$$= 0,908\%$$

Maka diperoleh data pada Tabel 5.11

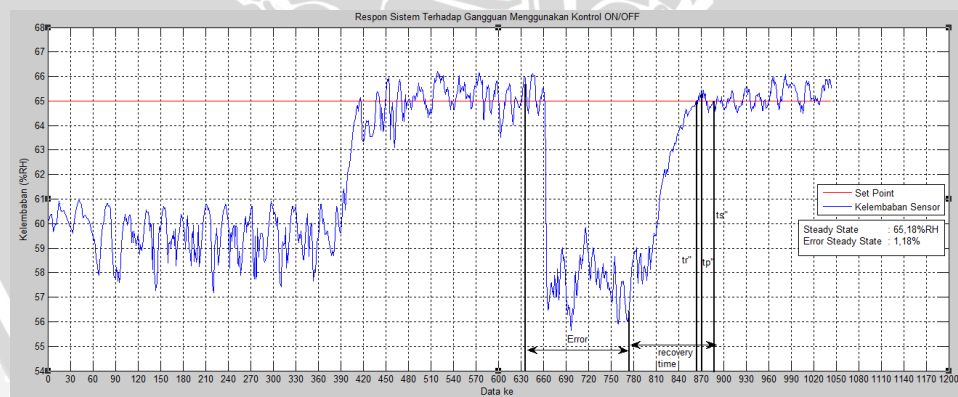
Tabel 5.11 Data Respon Kelembaban 60%RH Terhadap Gangguan Menggunakan Kontroler *On-Off*

Waktu Naik (t_r)	80 detik
Waktu Puncak (t_p)	90 detik
Waktu Keadaan Mantap (t_s)	190 detik
<i>Error Steady State</i>	0,908%

Keterangan:

- e_{ss} : *Error Steady State*
- $H_{\text{tengah steady state}}$: Kelembaban rata-rata dari data ke-787 sampai dengan data ke-~.
- $H_{\text{mula-mula}}$: Kelembaban saat data pertama atau waktu ke-0 detik
- H_{setpoint} : Kelembaban yang diinginkan, yaitu 60%RH
- Waktu Naik (t_r) adalah waktu yang diperlukan respons untuk naik dari 0-100% dari nilai akhirnya.
- Waktu Puncak (t_p) adalah waktu yang diperlukan respon untuk mencapai nilai puncak pertama respon.
- Waktu Keadaan Mantap (t_s) adalah waktu yang dibutuhkan respons untuk mencapai keadaan *steady state*.
- Data keluar setiap 2 detik.

Pengujian sistem kelembaban menggunakan kontroler *On-Off* dan *setpoint* kelembaban sebesar 65%RH dengan diberi gangguan berupa penutup mesin tetas dibuka selama 5 menit. Didapatkan hasil respon sistem sistem yang ditunjukkan pada Gambar 5.16



Gambar 5.16 Grafik Respon Kelembaban 65%RH Terhadap Gangguan Menggunakan Kontroler *On-Off*.

$$\begin{aligned}
 e_{ss} (\%) &= \left| \frac{(H_{\text{tengah steady state}}) - (H_{\text{setpoint}})}{(H_{\text{setpoint}}) - (H_{\text{mula-mula}})} \right| \times 100\% \\
 &= \left| \frac{65,18 - 65}{65 - 80,14} \right| \times 100\% \\
 &= 1,18\%
 \end{aligned}$$

Maka diperoleh data pada Tabel 5.12.

Tabel 5.12 Data Respon Kelembaban 65%RH Terhadap Gangguan Menggunakan Kontroler *On-Off*

Waktu Naik (t_r)	180 detik
Waktu Puncak (t_p)	190 detik
Waktu Keadaan Mantap (t_s)	220 detik
<i>Error Steady State</i>	1,18%

Keterangan:

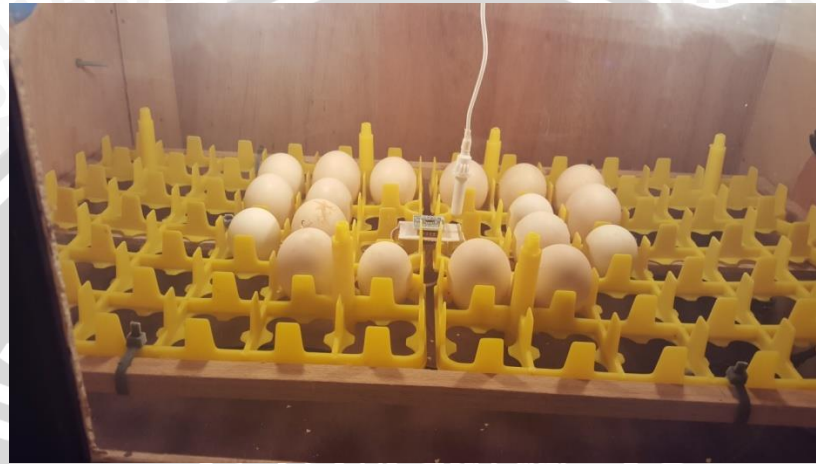
- e_{ss} : *Error Steady State*.
- $H_{\text{tengah steady state}}$: Kelembaban rata-rata dari data ke-887 sampai dengan data ke-~.
- $H_{\text{mula-mula}}$: Kelembaban saat data pertama atau waktu ke-0 detik.
- H_{setpoint} : Kelembaban yang diinginkan, yaitu 65%RH.
- Waktu Naik (t_r) adalah waktu yang diperlukan respons untuk naik dari 0-100% dari nilai akhirnya.
- Waktu Puncak (t_p) adalah waktu yang diperlukan respon untuk mencapai nilai puncak pertama respon.
- Waktu Keadaan Mantap (t_s) adalah waktu yang dibutuhkan respons untuk mencapai keadaan *steady state*.
- Data keluar setiap 2 detik.

Setelah melakukan pengujian sistem dan pengujian terhadap gangguan yang diberikan maka dari data pengujian menunjukkan bahwa sistem telah memenuhi kriteria mesin tetas yang diharapkan. Dari hasil pengujian diatas juga menunjukkan bahwa dengan menggunakan metode Kontrol Logika *Fuzzy* yang berbeda dan penambahan Kontroler *On-Off* pada aktuator kelembaban (*humidifier*) menghasilkan respon sistem yang lebih baik dari alat yang telah dibuat sebelumnya.

5.5 Hasil Penetasan

Penetasan menggunakan mesin tetas pada prinsipnya mengkondisikan telur ayam seperti telur ayam yang dierami oleh induknya. Penetasan pada telur ayam membutuhkan

waktu 21 hari sampai telur menetas dengan suhu 37,5°C, kelembaban 60% pada hari ke 1-18 dan 65%RH pada hari ke 19-21. Pada skripsi ini telur yang digunakan adalah telur ayam kampung tetas sebanyak 18 butir. Dikarenakan pada skripsi sebelumnya mendapatkan persentase penetasan sebesar 80% maka pada skripsi ini diharuskan persentase penetasan diatas 80%. Berikut adalah telur tetas yang sudah dimasukkan kedalam mesin tetas dan hasil penetasan selama 21 hari menggunakan mesin tetas. (lihat Gambar 5.17, 5.18, 5.19)



Gambar 5.17 Telur Tetas Didalam Mesin Tetas



Gambar 5.18 Hasil Penetasan



Gambar 5.19 Menetas Cacat

Maka diperoleh persentase penetasan sebesar:

$$\begin{aligned} \text{Persentase Penetasan (\%)} &= \frac{\text{telur menetas}}{\text{jumlah telur yang ditetaskan}} \times 100\% \\ &= \frac{16 \text{ butir}}{18 \text{ butir}} \times 100\% = 88,89\% \end{aligned}$$

Berikut adalah Tabel 5.13 hasil penetasan.

Tabel 5.13 Hasil penetasan

Jumlah Telur	Menetas	Tidak Menetas	Menetas Cacat
18	16	2	1

Dari hasil penetasan dapat disimpulkan bahwa penggunaan metode Kontrol Logika Fuzzy yang berbeda untuk suhu dan penggunaan kontroler On-Off untuk kelembaban meningkatkan persentase penetasan dari 80% menjadi 88,89%. Hal ini dikarenakan respon dari sistem suhu dan kelembaban yang lebih baik ketika diberikan gangguan maupun tidak, sehingga pengaturan suhu dan kelembaban ruang mesin tetap sesuai kebutuhan telur ayam saat proses penetasan.

BAB VI

PENUTUP

6.1 Kesimpulan

Berdasarkan pada perancangan alat dan hasil pengujian serta pengamatan yang telah dilakukan, dapat diambil beberapa kesimpulan, yaitu :

- 1) Penggunaan Kontrol Logika *Fuzzy* sebagai pengendali suhu menggunakan 3 *Membership Function* untuk masukan *error*, *delta error* dan keluaran berupa PWM dengan metode inferensi *Max-Min*, serta metode defuzzifikasi *Weighted Average*. Suhu dapat mencapai nilai setpoint 37,5°C dengan *Error steady state* kurang dari 2%.
- 2) Penggunaan Kontrol On-Off sebagai pengendali kelembaban. Kelembaban dapat mencapai nilai setpoint 60%RH dan 65%RH dengan *Error steady state* kurang dari 2%.
- 3) Telur yang ditetaskan sebanyak 18 butir. Diperoleh hasil 16 telur menetas dan 2 telur gagal menetas. Dari 16 telur yang menetas terdapat 1 anak ayam yang cacat.
- 4) Tujuan skripsi ini yaitu pengujian respon sistem suhu dan kelembaban harus lebih baik dari skripsi sebelumnya dan persentase penetasan lebih dari 80%. Dari hasil pengujian respon sistem dan penetasan diperoleh sistem dengan respon yang lebih baik ketika mendapatkan gangguan maupun tidak dan persentase penetasan sebesar 88,89% dimana persentase penetasan tersebut sudah memenuhi tujuan dari skripsi ini. (lihat Lampiran)

6.2 Saran

Dalam perancangan dan pembuatan alat ini masih terdapat kelemahan. Agar tercapainya penyempurnaan kinerja alat dan pengembangan lebih lanjut disarankan :

- Untuk penelitian lebih lanjut disarankan agar pengendalian kelembaban dapat menggunakan kontroler lain sehingga menghasilkan *error steady state* yang lebih kecil.
- Untuk memaksimalkan hasil dari penetasan sebaiknya jumlah telur dapat ditingkatkan sesuai dengan kapasitas mesin tetas.

DAFTAR PUSTAKA

Anonim, <http://id.wikihow.com/Menetaskan-Telur-Ayam>. (Diakses pada 1 Februari 2016)

Edhy, Sudjarwo, Komoditi Jenis Unggas Dari Tugas Akhir Penulis Selama Studi S-1(Skripsi) di FAPET UB (Ayam Ras), S-2 (Thesis) di FPS Ilmu Ternak Institut Pertanian Bogor (Puyuh) dan S-3 (Disertasi) di FPS Ilmu Ternak FAPET UB (Itik). Malang : Fakultas Peternakan UB, 2012. <http://edhysudjarwounggas.lecture.ub.ac.id/> (Diakses pada 12 Maret 2016)

Insani, Galuh Adi. 2010. *Beternak Unggas*. Jakarta: PT. Penebar Swadaya.

Nastiti, Rima. 2014. *Menjadi Milyader Budidaya Ayam Broiler*. Yogyakarta: Pustaka Baru Press.

Ogata, K. 1997. *Teknik Kontrol Automatik Jilid 2*. Jakarta: Penerbit Erlangga.

Paimin, Farry. 2000. *Membuat Dan Mengelola Mesin Tetas*. Penebar Swadaya. Jakarta.

Sensirion. 2002. SHT1x Relative Humidity & Temperature Sensor System. <http://www.sensirion.com>. Zurich: Sensirion. (Diakses 4 April 2016)

Sukardi, dkk. 1999. *Dasar Ternak Unggas*. Fakultas Peternakan UNSOED. Purwokerto.

Tri-Yuwanta. 1983. *Beberapa Metode Praktis Penetasan Telur*. Fakultas Peternakan UGM. Yogyakarta.

Riyanto, Anthonius. 2001. *Sukses Menetaskan Telur Ayam*. Jakarta : Agromedia Pustaka

LAMPIRAN

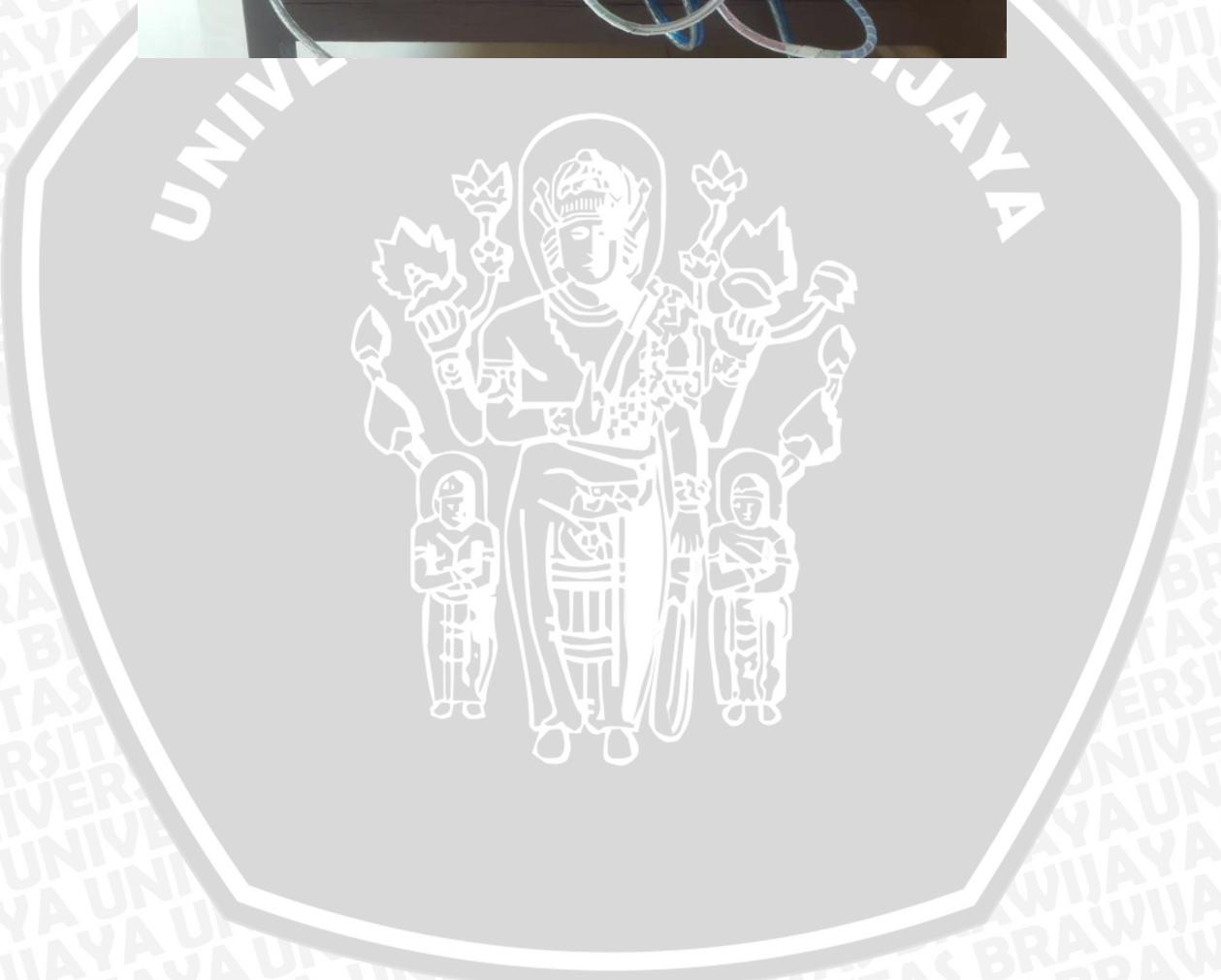
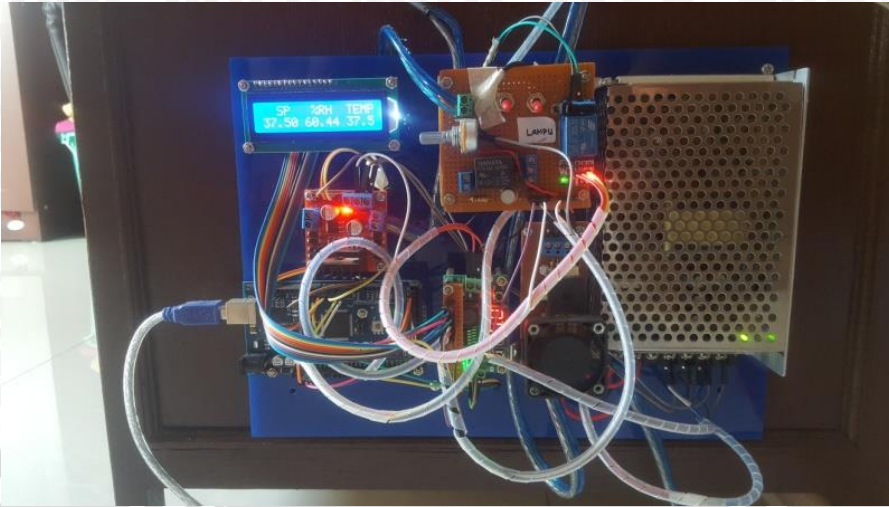
Tampak Keseluruhan



Tampak Dalam



Rangkaian Elektrik Brooder



Listing Program

```
#include<stdio.h>

#include<math.h>

#include "SHT1x.h"

#include <LiquidCrystal.h>

#define dataPin 9

#define clockPin 8

#define D 0

#define KP 1

#define I 2

#define P 3

#define PS 4

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

SHT1x sht1x(dataPin, clockPin);

//float getTemperatureRaw();

float humidity;

float temperature;

int AC_LOAD = 3;
```

```
int PWM; // PWM level (0-128) 0 = ON, 128 = OFF
```

```
float set_point; // temperatur set oint
```

```
float hset_point; // humidity set point
```

```
float herror;
```

```
float terror;
```

```
float derror;
```

```
float terror1;
```

```
float tError[6]; // jumlah array untuk error
```

```
float dError[6]; // jumlah array untuk delta error
```

```
float Dimmer_fuzzy[25]; // 5x5 MF untuk masukan
```

```
unsigned char rule_Dimmer[5][5]=
```

```
{
```

```
    {D,D,KP,KP,I}, // D = DINGIN
```

```
    {D,KP,KP,I,I}, // KP = KURANG PANAS
```

```
    {KP,KP,I,I,P}, // I = IDEAL
```

```
    {I,I,P,P,P}, // P = PANAS
```

```
    {P,P,PS,PS,PS}, // PS = PANAS SEKALI
```

```
};
```



```
float trapesium(float value, float x0, float x1, float x2, float x3);
```

```
float triangle(float value, float x0, float x1, float x2);
```

```
float reverse_grade(float value, float x0, float x1);
```

```
float grade(float value, float x0, float x1);
```

```
float fuzzyfikasi();
```

```
float rule_base_Dimmer();
```

```
float fuzzy_dimmer();
```

```
int relay = 5; //PIN humidifier
```

```
int enable = 10; // Pin enable Kipas
```

```
int in0 = 6; // Pin Driver Kipas
```

```
int in1 = 7; // Pin Driver Kipas
```

```
void setup()
```

```
{
```

```
Serial.begin(9600);
```

```
lcd.begin(16, 2);
```

```
pinMode(34, OUTPUT); //LCD
```

```
digitalWrite(34, HIGH); //LCD
```

```
pinMode(relay, OUTPUT); // Humidifier
```

```
digitalWrite(in0, HIGH); // Driver Kipas
```

```
digitalWrite(in1, LOW); // Driver Kipas
```

```
Serial.print("PWM");  
Serial.print("\t\t");  
Serial.print("set_point");  
Serial.print("\t");  
Serial.print("SUHU");  
Serial.print("\t");  
Serial.print("  error_suhu");  
Serial.print("\t");  
Serial.print("  error(t-1)");  
Serial.print("\t");  
Serial.print("  delta_error_suhu");  
Serial.print("\t");  
Serial.print("HMDTY");  
Serial.print("\t");  
Serial.print("  error_HMDTY");  
Serial.print("\n");  
  
set_point = 37.5;  
terror1=0;  
hset_point = 65;  
  
// :: SHT11
```

```
// pinMode(dataPin, OUTPUT);

//pinMode(clockPin, OUTPUT);

//digitalWrite(dataPin, HIGH);

//digitalWrite(clockPin, LOW);


// 3 :: Dimmer

pinMode(AC_LOAD, OUTPUT); // Set the AC Load as output
attachInterrupt(0, zero_crosss_int, RISING); // Choose the zero cross interrupt
}

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

void loop()
{
```



```
// 4 :: SHT11
```

```
// _val = getTemperatureRaw(); // Fetch raw value
```

```
// _temperature = _val * 0.01 + -44.15; // Convert raw value to degrees Celsius
```

```
temperature=sht1x.getTemperatureC();
```

```
humidity=sht1x.getHumidity();
```

```
terror = set_point - temperature; // mencari terror
```

```
derror = terror - terror1; // delta error
```

```
herror = hset_point - humidity; // mencari error kelembaban
```

```
PWM = fuzzy_dimmer();
```

```
terror1 = terror;
```

```
// Kontrol On-Off Humidifier
```

```
if (humidity<65) {
```

```
    digitalWrite(relay,HIGH);
```

```
}
```

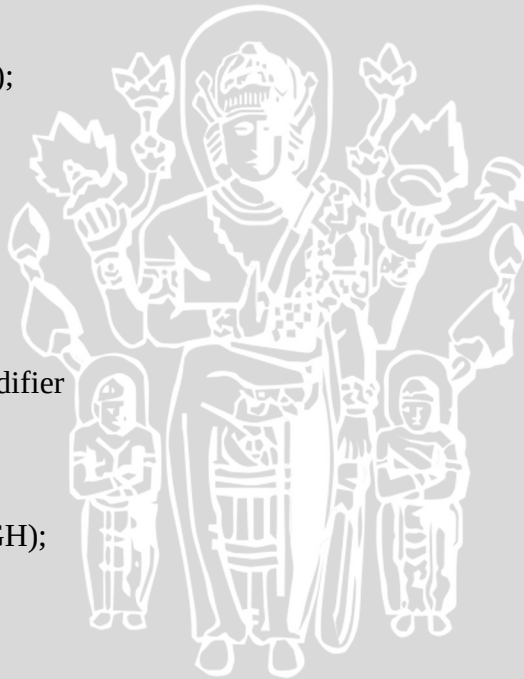
```
else {
```

```
    digitalWrite(relay,LOW);
```

```
}
```

```
// Kontrol On-Off Kipas
```

```
if (temperature>37.55) {
```



```
analogWrite(in0, 80);  
analogWrite(in1, LOW); }  
  
else {  
  
    analogWrite(in0, LOW);  
    analogWrite(in1, LOW);  
  
}
```

```
lcd.setCursor(2,0);  
lcd.print("SP");  
  
lcd.setCursor(0,1);  
lcd.print(set_point);  
  
lcd.setCursor(7,0);  
lcd.print("%RH");  
  
lcd.setCursor(6,2);  
lcd.print(humidity);  
  
lcd.setCursor(12,0);  
lcd.print("TEMP");  
  
lcd.setCursor(12,2);  
lcd.print(temperature);
```

```
Serial.print(PWM);  
  
Serial.print("\t\t");  
  
Serial.print(set_point);
```



```
Serial.print("\t\t");  
Serial.print(temperature);  
Serial.print("\t\t");  
Serial.print(terror);  
Serial.print("\t\t");  
Serial.print(terror1);  
Serial.print("\t\t");  
Serial.print(derror);  
Serial.print("\t\t");  
Serial.print(humidity);  
Serial.print("\t\t");  
Serial.print(herror);  
Serial.print("\n");  
  
delay(1000);  
}  
  
float fuzzy_dimmer()  
{  
    fuzzyfikasi();  
    return rule_base_Dimmer();  
}
```




```
/*=====
=====
```

```
*          FUNGSI FUZZYFIKASI
```

```
=====
=====*/
```

```
float fuzzyfikasi()
```

```
{
```

```
//KOMA pakai titik(.) misal 3,5 ditulis 3.5
```

```
tError[D] =reverse_grade(terror,-0.5,-0.25);
```

```
tError[KP] =triangle(terror,-0.5,-0.25,0);
```

```
tError[I] =triangle(terror,-0.25,0,0.25);
```

```
tError[P] =triangle(terror,0,0.25,0.5);
```

```
tError[PS] =grade(terror,0.25,0.5);
```

```
dError[D] =reverse_grade(derror,-0.5,-0.25);
```

```
dError[KP] =triangle(derror,-0.5,-0.25,0);
```

```
dError[I] =triangle(derror,-0.25,0,0.25);
```

```
dError[P] =triangle(derror,0,0.25,0.5);
```

```
dError[PS] =grade(derror,0.25,0.5);
```

```
/*=====
=====
```

*

FUNGSI CEK ATURAN FUZZY&DEFUZZYFIKASI

=====

=====*/

```
float rule_base_Dimmer()
```

```
{
```

```
    float umin;
```

```
    float umax; float Num,Denum; // Apa itu Num, Denum? pembilang penyebut
```

```
    float num[]={0,0,0,0,0}; //nilai awal array
```

```
    float denum[]={0,0,0,0,0}; //nilai awal array
```

```
    int x,y;
```

```
    float centre; //tengah
```

```
    float nilai_Dimmer;
```

```
    float Dimmer_out=0; // hasil defuzzyfikasi
```

```
    for(x=D;x<=PS;x++)
```

```
    {
```

```
        for(y=D;y<=PS;y++)
```

```
        {
```

```
            if(tError[x]>0 && dError[y]>0)
```

```
            {
```

```
                umin=(float)fmin(tError[x],dError[y]);
```



```
Dimmer_fuzzy[rule_Dimmer[x][y]]=(float)fmax(umin,Dimmer_fuzzy[rule_Dimmer[  
x][y]]);
```

```
if (rule_Dimmer[x][y]==0)
```

```
{
```

```
    centre=90;
```

```
}
```

```
else if(rule_Dimmer[x][y]==1)
```

```
{
```

```
    centre=70;
```

```
}
```

```
else if(rule_Dimmer[x][y]==2)
```

```
{
```

```
    centre=50;
```

```
}
```

```
else if(rule_Dimmer[x][y]==3)
```

```
{
```

```
    centre=30;
```

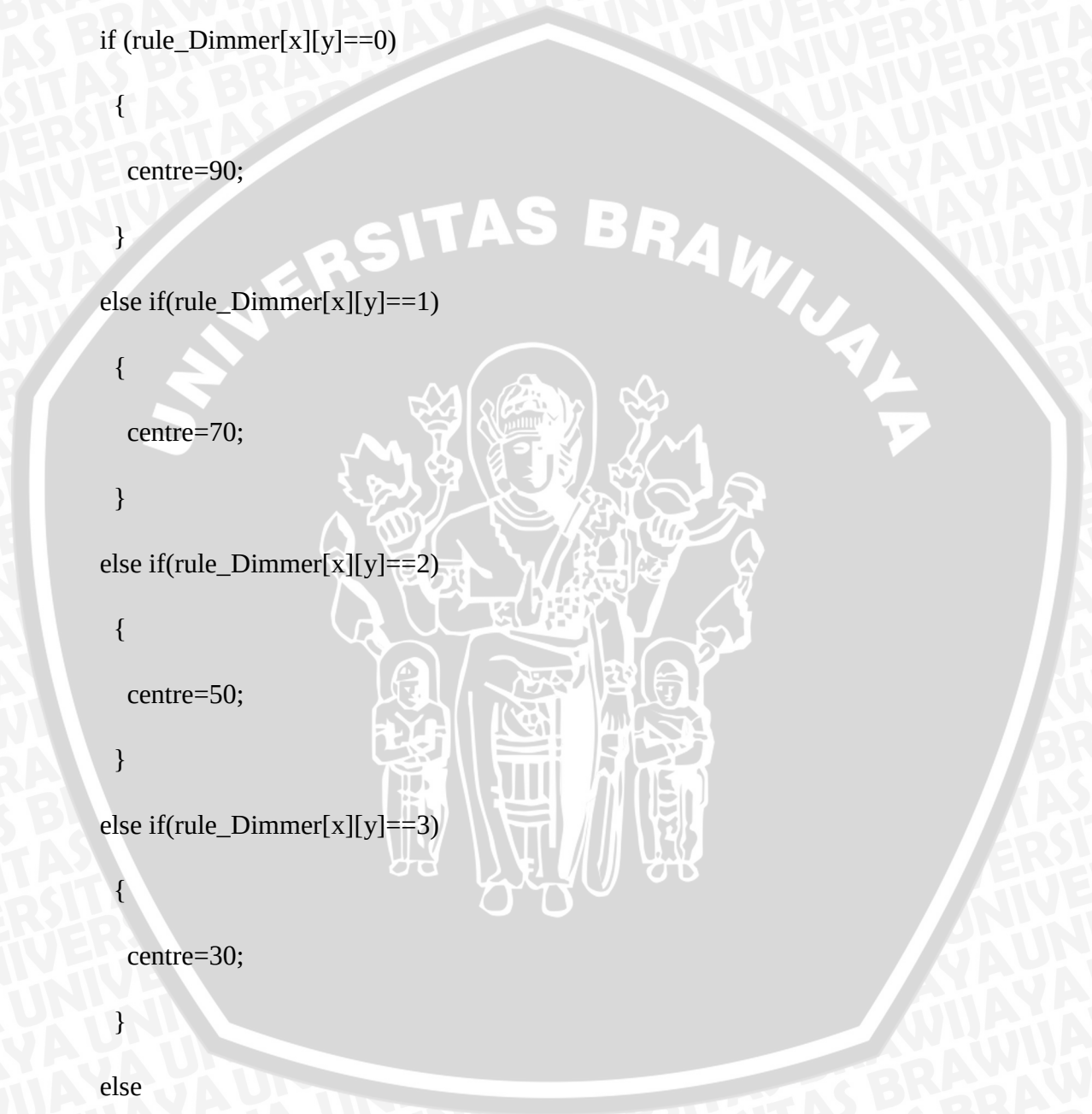
```
}
```

```
else
```

```
{
```

```
    centre=10;
```

```
}
```




```
num[rule_Dimmer[x][y]]=Dimmer_fuzzy[rule_Dimmer[x][y]]*centre;
```

```
denum[rule_Dimmer[x][y]]=Dimmer_fuzzy[rule_Dimmer[x][y]];
```

```
}
```

```
}
```

```
}
```

```
Num = num[0]+num[1]+num[2]+num[3]+num[4];
```

```
Denum = denum[0]+denum[1]+denum[2]+denum[3]+denum[4];
```

```
Dimmer_out=Num/Denum;
```

```
return Dimmer_out;
```

```
}
```

```
/*=====
```

```
=====
```

```
*
```

FUNGSI KEANGGOTAAN FUZZY

```
=====
```

```
=====*/
```

```
float trapesium(float 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-x0)/(x1-x0))+((x-x2)/(x3-x2)))/2);}

return result;
}
```

```
float triangle(float value, float x0, float x1, float x2)
```

```
{
    float result=0;

    float x;

    x=value;

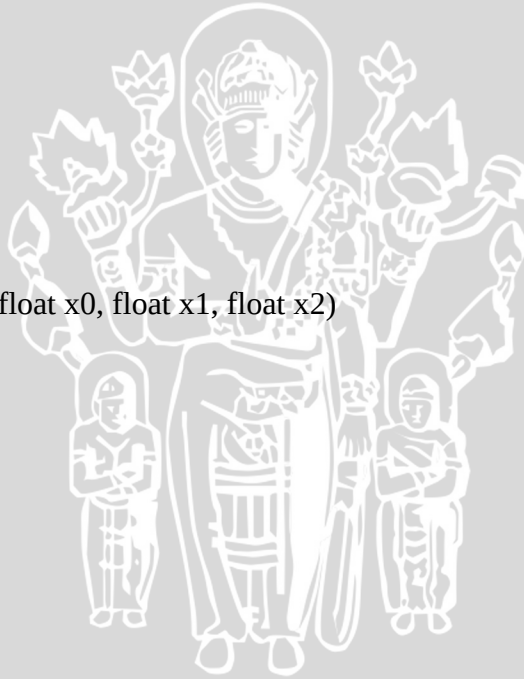
    if ((x<=x0)|| (x>=x2))

    {result=0;}

    else if(x==x1)

    {result=1;}

    else if((x>=x0)&&(x<x1))
```



```
{result=((x-x0)/(x1-x0));}
else result=(((x-x0)/(x1-x0)));
return result;
}
```

```
float grade(float 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(float value, float x0, float x1)
```

```
{
float result =0;
```

UNIVERSITAS BRAWIJAYA




```
float x;  
x=value;  
  
if(x<=x0)  
{result=1;}  
  
else if(x>=x1)  
{result=0;}  
  
else  
{result=(-x+x1)/(x1-x0);}  
  
return result;  
}
```



BAB VI KESIMPULAN

BAB VI
PENUTUP

A. Kesimpulan

Dari perancangan dan pembuatan sistem serta dilanjutkan dengan proses pengujian, maka diperoleh kesimpulan sebagai berikut :

1. Pada pengujian sistem pengendali suhu dan kelembaban pada mesin penetas telur ayam dengan set point suhu $38,3^{\circ}\text{C}$ dan kelembaban 70%RH, diketahui respon transien suhu dan kelembaban seperti dalam Tabel 6.1 dan Tabel 6.2.

Tabel 6.1 Respon Transien Suhu

Tanggapan transien suhu			
td (s)	tr (s)	ts (s)	Ess (%)
270	450	920	0,55

Tabel 6.2 Respon Transien Kelembaban

Tanggapan transien kelembaban			
td (s)	tr (s)	ts (s)	Ess (%)
140	140	240	4,28

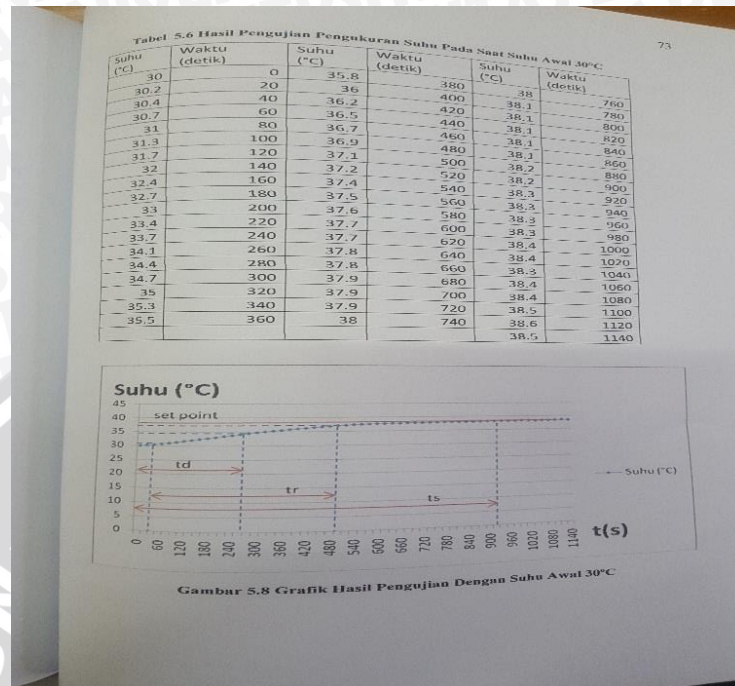
2. Pada pengujian pengukuran sistem pengendali suhu dan kelembaban pada mesin penetas telur ayam dengan set point suhu $38,3^{\circ}\text{C}$ dan kelembaban 70%RH saat diberi gangguan, diketahui *recovery time* suhu selama 520 detik dan *recovery time* kelembaban selama 60 detik
3. Dari pengujian alat dengan telur ayam, dapat diketahui bahwa keberhasilan mesin penetas telur ini adalah 80%.

B. Saran

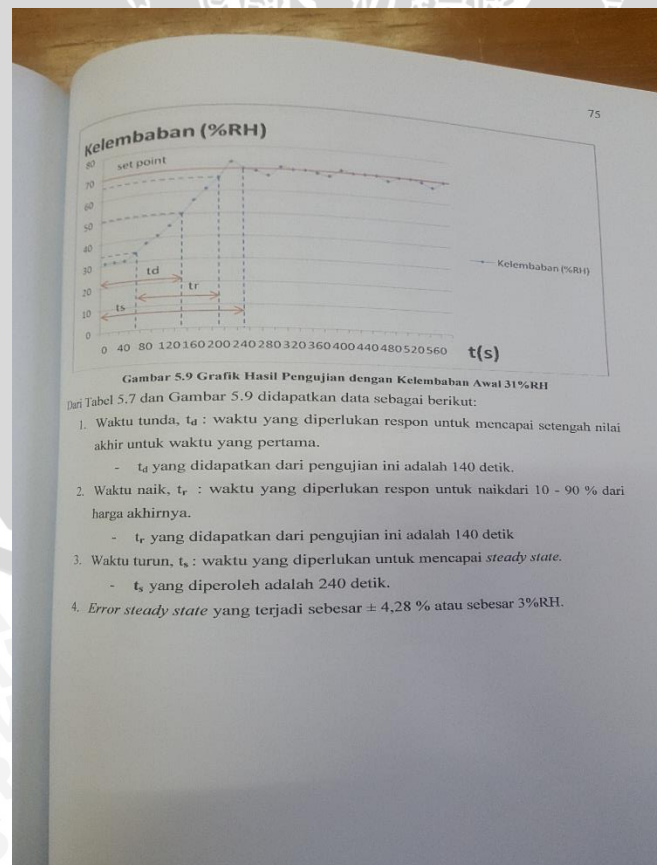
Dalam perancangan dan pembuatan alat ini masih terdapat kelemahan dan kekurangan sistem. Beberapa hal yang dapat dijadikan saran dalam alat ini adalah:

1. Pada skripsi ini hanya dapat mengontrol telur sebanyak 6-18 butir saja, maka dalam perkembangannya dapat diaplikasikan untuk keperluan yang lebih besar.
2. Perlu adanya kajian yang lebih lengkap untuk hubungan elemen pemanas dengan volume atau ruang udara tertentu serta besarnya ruang ventilasi.
3. Alat penetas telur ayam ini dapat diaplikasikan pada berbagai varietas telur lain seperti telur itik, telur puyuh, telur merpati, dll dengan memperhatikan

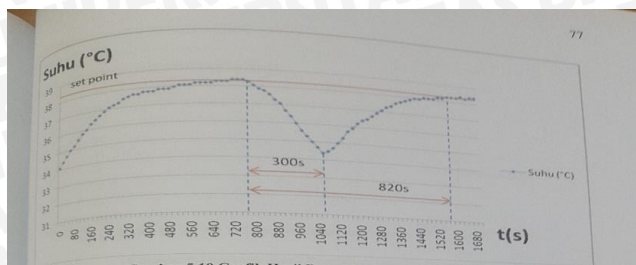
SUHU



KELEMBABAN



GANGGUAN SUHU



Gambar 5.10 Grafik Hasil Pengujian dengan Gangguan Pada Saat Suhu Awal 34,1°C

Dari Tabel 5.8 dan Gambar 5.10 didapatkan data sebagai berikut:

Suhu dalam ruangan dapat berubah karena pengaruh suhu luar. Gangguan diberikan dengan cara pintu prototipe ruangan dibuka selama 300 detik.

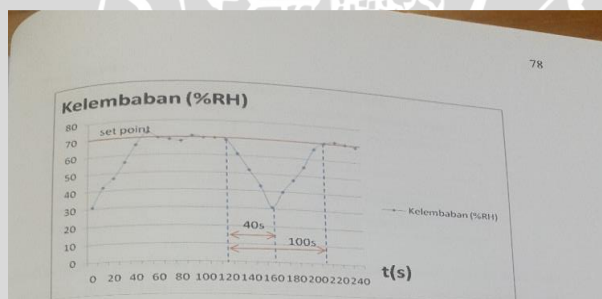
-Pintu dibuka pada saat mencapai steady state, lebih tepatnya pada detik 760.

-Setelah 300 detik, pintu ditutup tepatnya pada detik ke 1060. Dan didapatkan suhu sebesar 34,4°C sehingga terjadi error *steady state* sebesar: $\pm 10,18\%$ atau sebesar 3,9°C dan kembali mencapai *steady state* setelah 520 detik pintu ditutup.

Tabel 5.9 Hasil Pengujian Dengan Gangguan Pada Saat Kelembaban Awal 31%RH

Kelembaban (%RH)	Waktu (detik)	Kelembaban (%RH)	Waktu (detik)
31	0	69	120
42	10	61	130
47	20	52	140
56	30	43	150
66	40	31	160
73	50	40	170
70	60	47	180
69	70	55	190
68	80	66	200
71	90	70	210
70	100	71	220
70	110	70	230
		69	240

GANGGUAN KELEMBABAN



Gambar 5.11 Grafik Hasil Pengujian dengan Gangguan Pada Saat Kelembaban Awal 31%RH

Dari Tabel 5.9 dan Gambar 5.11 didapatkan data sebagai berikut:

Kelembaban dalam ruangan dapat berubah karena pengaruh dari luar. Gangguan diberikan dengan cara pintu prototipe ruangan dibuka selama 40 detik.

Pintu dibuka pada saat mencapai steady state, lebih tepatnya pada detik 120.

-Setelah 40 detik, pintu ditutup tepatnya pada detik ke 160. Dan didapatkan kelembaban sebesar 31 %RH sehingga terjadi error *steady state* sebesar: $\pm 55,7\%$ atau sebesar 39°C dan kembali mencapai *steady state* setelah 60 detik pintu ditutup.

Salah satu contoh pengujian penetasan yang dilakukan terhadap telur ayam diberikan dalam Tabel 5.10 dan 5.11

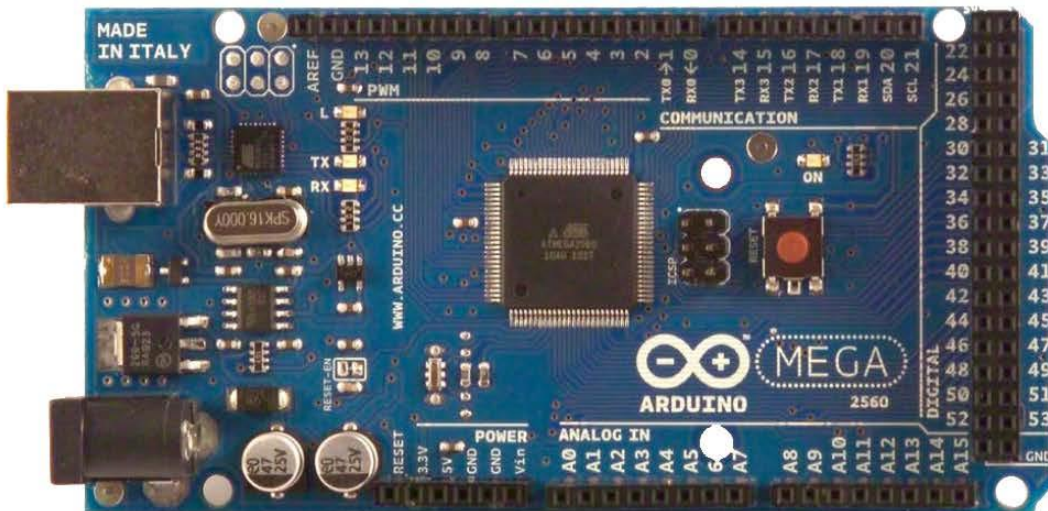
Pengujian I

Jumlah telur yang ditetaskan sebanyak 15 butir telur.

Tabel 5.10 Pengujian I Keberhasilan Penetasan Telur

Hari ke-	Jumlah telur yang menetas
18	-
19	1 telur menetas
20	2 telur menetas
21	9 telur menetas

Arduino MEGA 2560



Product Overview

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 ([datasheet](#)). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

Index

Technical
Specifications

Page 2

How to use Arduino
Programming Enviroment, Basic Tutorials

Page 6

Terms &
Conditions

Page 7

Enviromental Policies
half sqm of green via Impatto Zero®

Page 7



radiospares

RADIONICS



Technical Specification

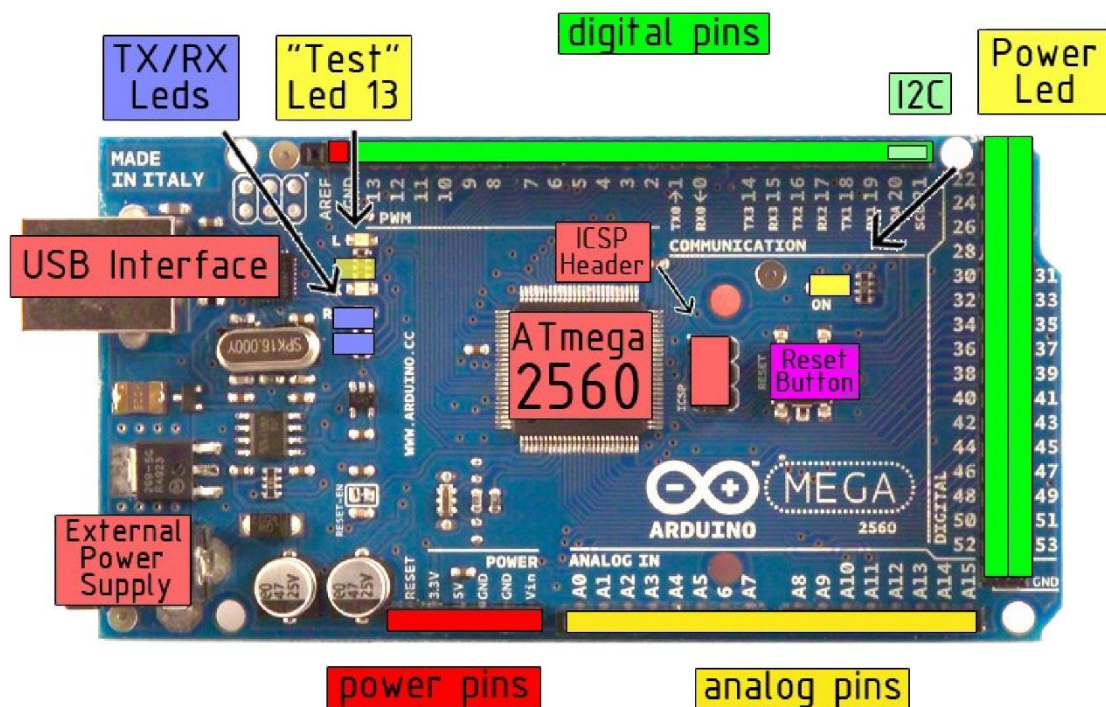


EAGLE files: [arduino-mega2560-reference-design.zip](#) Schematic: [arduino-mega2560-schematic.pdf](#)

Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

the board



radiospares

RADIONICS



Power

The Arduino Mega2560 can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2).** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM: 0 to 13.** Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- **SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Duemilanove and Diecimila.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- **I²C: 20 (SDA) and 21 (SCL).** Support I²C (TWI) communication using the [Wire library](#) (documentation on the Wiring website). Note that these pins are not in the same location as the I²C pins on the Duemilanove.

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and [analogReference\(\)](#) function.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with [analogReference\(\)](#).
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.



radiospares

RADIONICS



Communication

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically). The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Mega's digital pins.

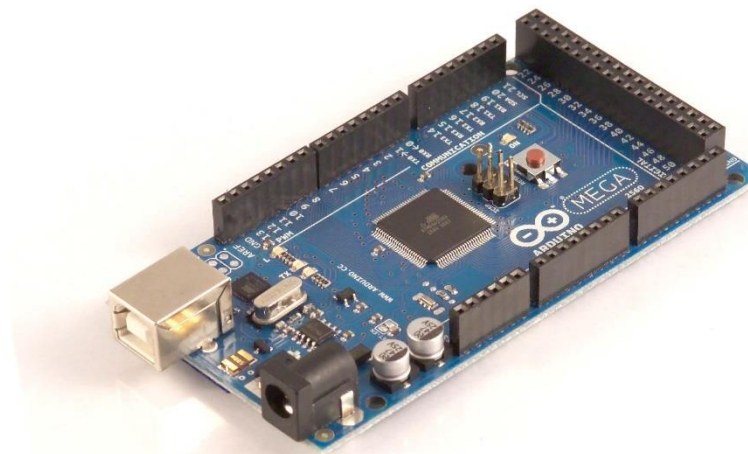
The ATmega2560 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the [documentation on the Wiring website](#) for details. To use the SPI communication, please see the ATmega2560 datasheet.

Programming

The Arduino Mega2560 can be programmed with the Arduino software ([download](#)). For details, see the [reference](#) and [tutorials](#).

The ATmega2560 on the Arduino Mega comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.



radiospares

RADIONICS



Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Mega2560 is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega2560 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Mega2560 is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Mega2560. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Mega contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

USB Overcurrent Protection

The Arduino Mega has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics and Shield Compatibility

The maximum length and width of the Mega PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

The Mega is designed to be compatible with most shields designed for the Diecimila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega and Duemilanove / Diecimila. **Please note that I²C is not located on the same pins on the Mega (20 and 21) as the Duemilanove / Diecimila (analog inputs 4 and 5).**



radiospares

RADIONICS



How to use Arduino



Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the [Arduino programming language](#) (based on [Wiring](#)) and the Arduino development environment (based on [Processing](#)). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

Arduino is a cross-platoform program. You'll have to follow different instructions for your personal OS. Check on the [Arduino site](#) for the latest instructions. <http://arduino.cc/en/Guide/HomePage>

Linux Install

Windows Install

Mac Install

Once you have downloaded/unzipped the arduino IDE, you can Plug the Arduino to your PC via USB cable.

Blink led

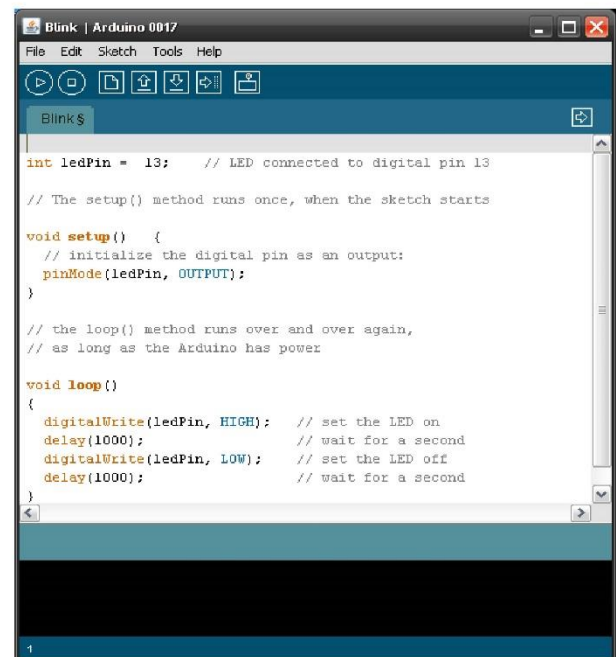
Now you're actually ready to "burn" your first program on the arduino board. To select "blink led", the physical translation of the well known programming "hello world", select

**File>Sketchbook>
Arduino-0017>Examples>
Digital>Blink**

Once you have your skecth you'll see something very close to the screenshot on the right.

In **Tools>Board** select MEGA

Now you have to go to
Tools>SerialPort
and select the right serial port, the one arduino is attached to.



Done compiling.

Press Compile button
(to check for errors)



Upload



TX RX Flashing



Blinking Led!



radiospares

RADIONICS



ALLIED ELECTRONICS
 800-541-4373
 www.alliedelec.com

Terms & Conditions



1. Warranties

1.1 The producer warrants that its products will conform to the Specifications. This warranty lasts for one (1) years from the date of the sale. The producer shall not be liable for any defects that are caused by neglect, misuse or mistreatment by the Customer, including improper installation or testing, or for any products that have been altered or modified in any way by a Customer. Moreover, The producer shall not be liable for any defects that result from Customer's design, specifications or instructions for such products. Testing and other quality control techniques are used to the extent the producer deems necessary.

1.2 If any products fail to conform to the warranty set forth above, the producer's sole liability shall be to replace such products. The producer's liability shall be limited to products that are determined by the producer not to conform to such warranty. If the producer elects to replace such products, the producer shall have a reasonable time to replacements. Replaced products shall be warranted for a new full warranty period.

1.3 EXCEPT AS SET FORTH ABOVE, PRODUCTS ARE PROVIDED "AS IS" AND "WITH ALL FAULTS." THE PRODUCER DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE

1.4 Customer agrees that prior to using any systems that include the producer products, Customer will test such systems and the functionality of the products as used in such systems. The producer may provide technical, applications or design advice, quality characterization, reliability data or other services. Customer acknowledges and agrees that providing these services shall not expand or otherwise alter the producer's warranties, as set forth above, and no additional obligations or liabilities shall arise from the producer providing such services.

1.5 The Arduino™ products are not authorized for use in safety-critical applications where a failure of the product would reasonably be expected to cause severe personal injury or death. Safety-Critical Applications include, without limitation, life support devices and systems, equipment or systems for the operation of nuclear facilities and weapons systems. Arduino™ products are neither designed nor intended for use in military or aerospace applications or environments and for automotive applications or environment. Customer acknowledges and agrees that any such use of Arduino™ products which is solely at the Customer's risk, and that Customer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

1.6 Customer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products and any use of Arduino™ products in Customer's applications, notwithstanding any applications-related information or support that may be provided by the producer.

2. Indemnification

The Customer acknowledges and agrees to defend, indemnify and hold harmless the producer from and against any and all third-party losses, damages, liabilities and expenses it incurs to the extent directly caused by: (i) an actual breach by a Customer of the representation and warranties made under this terms and conditions or (ii) the gross negligence or willful misconduct by the Customer.

3. Consequential Damages Waiver

In no event the producer shall be liable to the Customer or any third parties for any special, collateral, indirect, punitive, incidental, consequential or exemplary damages in connection with or arising out of the products provided hereunder, regardless of whether the producer has been advised of the possibility of such damages. This section will survive the termination of the warranty period.

4. Changes to specifications

The producer may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." The producer reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.



Enviromental Policies



The producer of Arduino™ has joined the Impatto Zero® policy of LifeGate.it. For each Arduino board produced is created / looked after half squared Km of Costa Rica's forest's.



radiospares

RADIONICS



DT-SENSE SHT11 MODULE

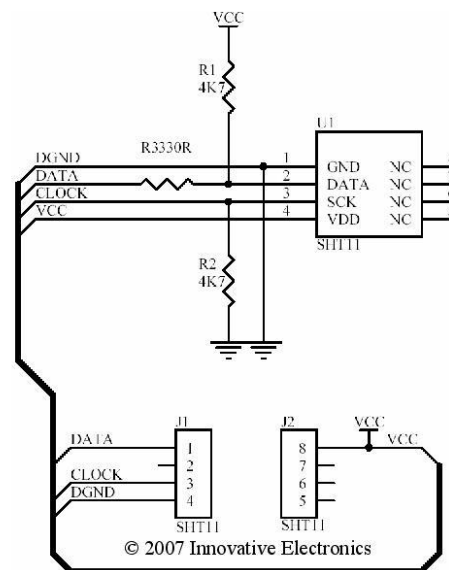
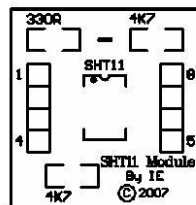
DT-Sense SHT11 Module merupakan suatu modul sensor suhu dan kelembaban yang berbasis Sensirion SHT11 (www.sensirion.com) dengan antarmuka Two-Wire Serial Interface. Modul ini dapat diaplikasikan dalam sistem pengendali suhu dalam ruang atau sistem weather station.

Spesifikasi Hardware

1. Range suhu : -40°C (-40°F) hingga +123,8°C (+254,9°F)
2. Akurasi suhu : +/- 0,5°C pada 25°C
3. Range kelembaban : 0 hingga 100% RH
4. Akurasi RH absolut : +/- 3,5% RH
5. Faktor bentuk : 8 pin DIP – 0,6"
6. Konsumsi daya rendah (tipikal 30 µV)
7. Tegangan supply +5 VDC

Tata Letak

Pin	Fungsi
1	Data
3	Clock
4	Ground
8	+5 VDC



Aplikasi terdapat di www.innovativeelectronics.com.

- ♦ Terima Kasih atas kepercayaan Anda menggunakan produk kami, bila ada kesulitan, pertanyaan atau saran mengenai produk ini silahkan menghubungi technical support kami :

support@innovativeelectronics.com

DT-SENSE SHT11 MODULE

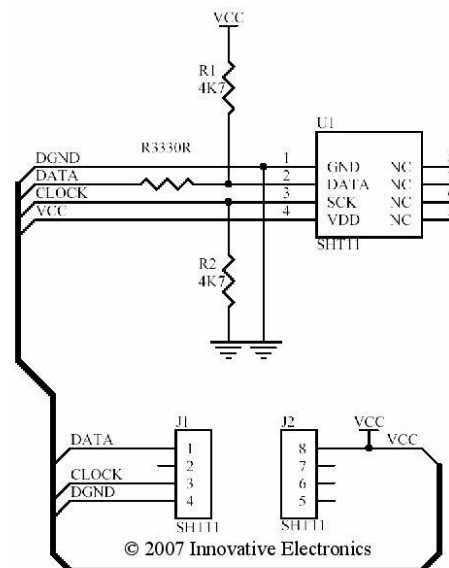
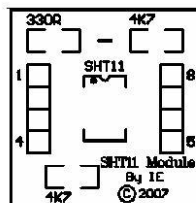
DT-Sense SHT11 Module merupakan suatu modul sensor suhu dan kelembaban yang berbasis Sensirion SHT11 (www.sensirion.com) dengan antarmuka Two-Wire Serial Interface. Modul ini dapat diaplikasikan dalam sistem pengendali suhu dalam ruang atau sistem weather station.

Spesifikasi Hardware

1. Range suhu : -40°C (-40°F) hingga +123,8°C (+254,9°F)
2. Akurasi suhu : +/- 0,5°C pada 25°C
3. Range kelembaban : 0 hingga 100% RH
4. Akurasi RH absolut : +/- 3,5% RH
5. Faktor bentuk : 8 pin DIP – 0,6"
6. Konsumsi daya rendah (tipikal 30 µV)
7. Tegangan supply +5 VDC

Tata Letak

Pin	Fungsi
1	Data
3	Clock
4	Ground
8	+5 VDC



Aplikasi terdapat di www.innovativeelectronics.com.

- ♦ Terima Kasih atas kepercayaan Anda menggunakan produk kami, bila ada kesulitan, pertanyaan atau saran mengenai produk ini silahkan menghubungi technical support kami :

support@innovativeelectronics.com