

**PENGENDALIAN SUHU DAN KELEMBABAN PADA BUDIDAYA
JAMUR TIRAM DENGAN MENGGUNAKAN METODE KONTROL
LOGIKA FUZZY**

SKRIPSI

TEKNIK ELEKTRO KONSENTRASI SISTEM KONTROL

Ditujukan untuk Memenuhi Persyaratan
Memperoleh Gelar Sarjana Teknik



ADE BARLIAN TANDIONO
NIM. 0910630001- 63

**KEMENTERIAN RISET TEKNOLOGI DAN PENDIDIKAN TINGGI
UNIVERSITAS BRAWIJAYA
FAKULTAS TEKNIK
JURUSAN TEKNIK ELEKTRO
MALANG
2016**

LEMBAR PERSETUJUAN

**PENGENDALIAN SUHU DAN KELEMBABAN PADA BUDIDAYA
JAMUR TIRAM DENGAN MENGGUNAKAN METODE KONTROL
LOGIKA FUZZY**

SKRIPSI

Digunakan Untuk Memenuhi Persyaratan
Memperoleh Gelar Sarjana Teknik



Disusun Oleh :

ADE BARLIAN TANDIONO

NIM. 0910633001 – 63

Telah diperiksa dan disetujui oleh :

Dosen Pembimbing I

Dosen Pembimbing II

M. Aziz Muslim, ST., MT., PhD.
NIP. 19741203 200012 1 001

Ir., Dipl., Ing., Moch. Rusli
NIP. 19630104 1987011 001

PERNYATAAN ORISINILITAS 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 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 ini dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, pasal 25 ayat 2, dan pasal 70)

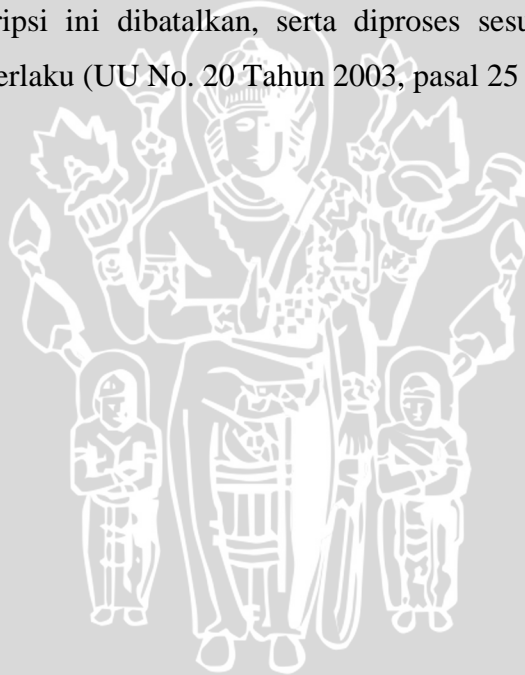
Malang, 26 Januari 2016

Mahasiswa,

Materai Rp. 6000,-

Ade Barlian Tandiono

0910630001





PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa yang telah memberikan kasih dan berkat-Nya yang tak berkesudahan, sehingga penulis dapat menyelesaikan skripsi yang berjudul “Pengendalian Suhu dan Kelembaban Pada Budidaya Jamur Tiram Dengan Menggunakan Metode Kontrol Logika Fuzzy”. Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik di Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya.

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

- Keluarga tercinta, kedua orang tua yaitu Papa tercinta Pratiktio Tandiono, Mama tercinta VS Winarni, selaku keluarga besar penulis yang selalu memberikan dukungan, semangat, nasihat, kasih sayang dan inspirasi terus menerus dengan sabar.
- Bapak M. Aziz Muslim, ST., MT, Ph.D sebagai Ketua Jurusan Teknik Elektro Universitas Brawijaya.
- Bapak Hadi Suyono, ST., MT, Ph.D sebagai Sekretaris Jurusan Teknik Elektro Universitas Brawijaya.
- Bapak Ali Mustofa ST., MT., selaku Ketua Program Studi S1 Teknik Elektro.
- Bapak Ir. Purwanto, MT. sebagai Ketua Kelompok Dosen Keahlian Sistem Kontrol Jurusan Teknik Elektro Universitas Brawijaya.
- Bapak M. Aziz Muslim, ST., MT, Ph.D. sebagai Dosen Pembimbing I atas segala bimbingan, pengarahan, saran, kritik dan masukan yang telah diberikan.
- Bapak Ir., Dipl., Ing., Moch. Rusli sebagai Dosen Pembimbing II atas segala bimbingan, pengarahan, gagasan, ide, saran serta motivasi yang telah diberikan.
- Staf Rekording, staf Pengajaran, dan staf Ruang Baca Jurusan Teknik Elektro yang telah membantu segala urusan penulis selama ini.

- Semua anggota Ampere 2009, teman-teman Elektro dari berbagai angkatan dan teman-teman Fakultas Teknik lainnya atas seluruh bantuan dan dukungannya.

Seluruh teman - teman serta semua pihak yang tidak mungkin untuk dicantumkan namanya satu-persatu, terima kasih banyak atas segala bentuk bantuan dan dukungannya.

Dalam penyusunan skripsi ini, penulis menyadari bahwa skripsi ini belumlah sempurna, karena keterbatasan ilmu dan kendala-kendala lain yang terjadi selama pengerjaan skripsi ini. Oleh karena itu, penulis mengharap kritik dan saran untuk penyempurnaan tulisan di masa yang akan datang. Penulis berharap, semoga tulisan ini dapat bermanfaat bagi kita semua.

Januari 2016

Penulis

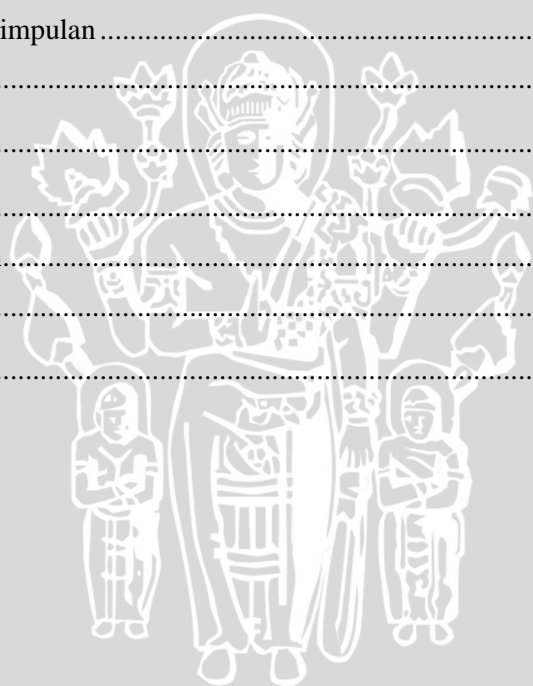


DAFTAR ISI

PENGANTAR	i
DAFTAR ISI.....	iii
DAFTAR GAMBAR	vi
DAFTAR TABEL.....	viii
RINGKASAN	ix
BAB I.....	1
PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Batasan Masalah.....	2
1.4. Tujuan	2
1.5. Sistematika Pembahasan.....	2
BAB II.....	4
DASAR TEORI	4
2.1. Fuzzy Logic Control.....	4
2.1.1. Struktur Dasar Kontrol Logika <i>Fuzzy</i>	5
2.1.2. Fungsi Keanggotaan	5
2.1.3. Fuzzy Logic Controller.....	7
2.1.4. Fuzzifikasi	7
2.1.5. Kaidah Aturan Fuzzy (Fuzzy Rule).....	8
2.1.6. Metode Inferensi MAX-MIN.....	9
2.1.7. Metode Defuzzifikasi Center Of Gravity (COG).....	10
2.2. PWM (Pulse Width Modulation).....	10
2.3. Arduino Uno.....	11
2.3.1. Daya	12
2.3.2. Memori	13
2.3.3. Input dan Output.....	13
2.3.4. Komunikasi	14
2.4. SHT 11	15

2.4.1.	Pengertian	15
2.4.2.	Prinsip Kerja Sensor SHT 11	15
2.5.	Motor DC	16
2.6.	Jamur Tiram	18
BAB III		20
METODOLOGI PENELITIAN		20
3.1.	Penentuan Spesifikasi Alat	20
3.2.	Perancangan Sistem	20
3.3.	Realisasi Pembuatan Sistem	21
3.3.1.	Perangkat Keras	21
3.3.2.	Perangkat Lunak	21
3.4.	Pengujian Alat	21
3.5.	Pengambilan Kesimpulan dan Saran	21
BAB IV		22
PERANCANGAN & PEMBUATAN ALAT		22
4.1.	Diagram Blok Sistem	22
4.2.	Prinsip Kerja Alat	23
4.3.	Perancangan Perangkat Keras	23
4.3.1.	Spesifikasi Alat	23
4.3.3.	Rangkaian <i>Driver</i> Motor DC	25
4.3.4.	Liquid Crystal Display (LCD)	26
4.3.5.	Modul Arduino Uno Rev3	28
4.4.	Perancangan Sistem Kontrol Logika <i>Fuzzy</i>	30
4.4.1.	Variabel Masukan dan Keluaran	30
4.4.2.	Fungsi Keanggotaan Masukan	30
4.4.3.	Fungsi Keanggotaan Keluaran	32
4.4.4.	Perancangan Aturan <i>Fuzzy</i>	33
4.4.5.	Inferensi Min-Max	34
4.4.6.	Defuzzifikasi	40
4.5.	Perancangan Perangkat Lunak	43
BAB V		44

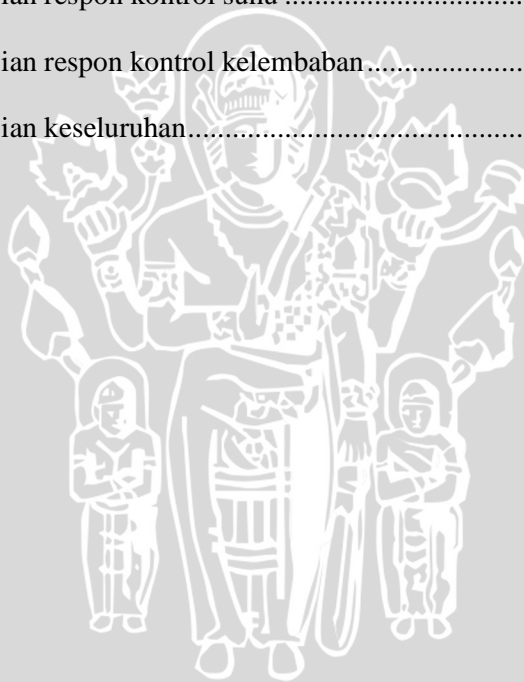
PENGUJIAN & ANALISIS SISTEM.....	44
5.1. Pengujian Rangkain Sensor Suhu dan Kelembaban SHT11	44
5.1.1. Prosedur Pengujian.....	44
5.1.2. Hasil Data Pengujian	45
5.1.3. Analisa dan Kesimpulan	46
5.2. Pengujian Respon alat	47
5.2.1. Pengujian Respon Kontrol Suhu.....	47
5.2.2. Pengujian Respon Kontrol Kelembaban	49
5.3. Pengujian Alat Keseluruhan.....	51
5.3.1. Prosedur Percobaan	51
5.3.2. Hasil Percobaan	52
5.3.3. Analisa dan Kesimpulan	53
BAB VI.....	54
PENUTUP	54
6.1. Kesimpulan	54
6.2. Saran Pengembangan.....	54
DAFTAR PUSTAKA	55
LAMPIRAN.....	56



DAFTAR GAMBAR

Gambar 2.1 Pengendali Fuzzy	5
Gambar 2.2 Fungsi Keanggotaan Bentuk Triangular	6
Gambar 2.3 Fungsi Keanggotaan Bentuk Trapesium	7
Gambar 2.4 Inferensi Fuzzy dengan Metode MAX-MIN	9
Gambar 2.5 Sinyal PWM Secara Umum	11
Gambar 2.6 Tampak depan Arduino Uno	12
Gambar 2.7 Sensirion Temperature/Humidity Sensor - SHT11	15
Gambar 2.8 Motor DC	17
Gambar 4.1 Blok Diagram Sistem	22
Gambar 4.2 Mekanik Miniatur Kumbung Jamur	24
Gambar 4.3 Bentuk fisik SHT yang berupa : (a) Modul (b) Skematik [6]	25
Gambar 4.4 SHT11 drawing and footprint dimensions in mm (inch) [6]	25
Gambar 4.5 Modul Rangkaian <i>Driver</i> EMS 30A <i>H-Bridge</i>	26
Gambar 4.6 LCD ke Arduino Uno	27
Gambar 4.7 Rangkaian Wiring LCD ke Arduino Uno	27
Gambar 4.8 Desain Sistem Arduino Uno Rev3	28
Gambar 4.9 Modul Arduino Uno	29
Gambar 4.10 Fungsi Keanggotaan Masukan <i>Error</i> dan <i>Delta Error Suhu</i>	31
Gambar 4.11 Fungsi Keanggotaan Masukan <i>Error</i> dan <i>Delta Error Kelembaban</i>	31
Gambar 4.12 Fungsi Keanggotaan Keluaran <i>Error</i> dan <i>Delta Error Suhu</i>	32
Gambar 4.13 Fungsi Keanggotaan Keluaran <i>Error</i> dan <i>Delta Error Kelembaban</i>	33
Gambar 4.14 Inferensi Fuzzy dengan Metode MIN-MAX	34
Gambar 4.15 Inferensi Min-Max data suhu ke-1	35
Gambar 4.16 Inferensi Min-Max data kelembaban ke-1	36

Gambar 4.17 Inferensi Min-Max data suhu ke-4.....	36
Gambar 4.18 Inferensi Min-Max data kelembaban ke-4	37
Gambar 4.19 Inferensi Min-Max data suhu ke-5.....	38
Gambar 4.20 Inferensi Min-Max data kelembaban ke-5	39
Gambar 4.21 <i>Flowchart</i> Program.....	43
Gambar 5.1 Pengujian Akurasi Sensor Kelembaban SHT11	45
Gambar 5.2 Grafik hasil pengujian sensor suhu <i>SHT11</i>	46
Gambar 5.3 Grafik hasil pengujian sensor kelembaban <i>SHT11</i>	46
Gambar 5.4 Grafik hasil pengujian respon kontrol suhu	48
Gambar 5.5 Grafik hasil pengujian respon kontrol kelembaban.....	50
Gambar 5.6 Grafik hasil pengujian keseluruhan.....	52



DAFTAR TABEL

Tabel 4.1 Fungsi Pin Arduino Uno Rev329

Tabel 4.2 Aturan Fuzzy (pengaturan kipas dan pompa).....33

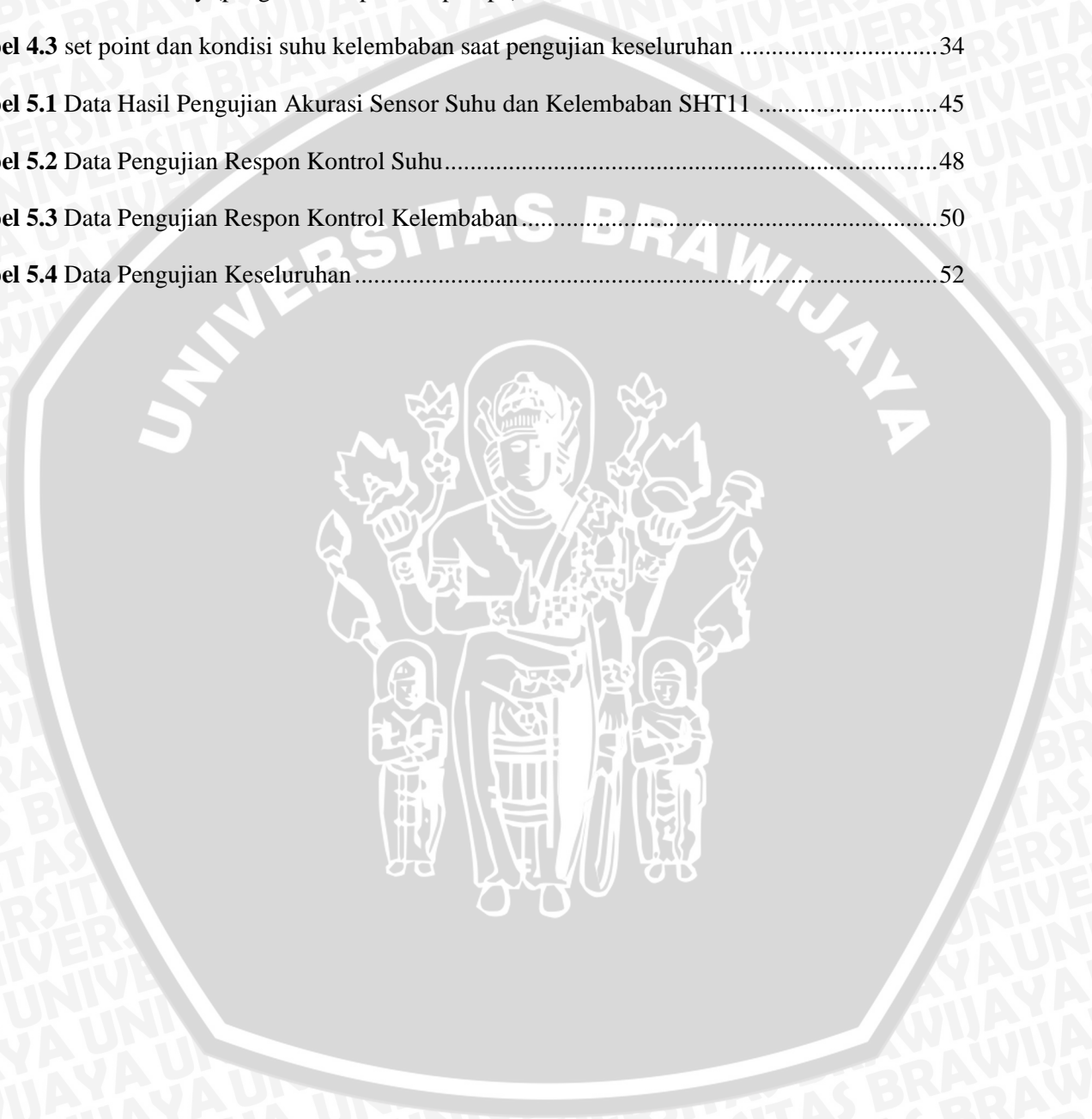
Tabel 4.3 set point dan kondisi suhu kelembaban saat pengujian keseluruhan34

Tabel 5.1 Data Hasil Pengujian Akurasi Sensor Suhu dan Kelembaban SHT1145

Tabel 5.2 Data Pengujian Respon Kontrol Suhu.....48

Tabel 5.3 Data Pengujian Respon Kontrol Kelembaban50

Tabel 5.4 Data Pengujian Keseluruhan52



RINGKASAN

Ade Barlian Tandiono, Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya, Januari 2016, *Pengendalian Suhu dan Kelembaban Pada Budidaya Jamur Tiram Dengan Menggunakan Metode Kontrol Logika Fuzzy*, Dosen Pembimbing : M. Aziz Muslim dan Moch. Rusli.

Perkembangan budidaya jamur tiram saat ini mengalami pertumbuhan yang sangat pesat ditandai dengan semakin banyaknya petani pembibit maupun pembesaran jamur yang berbanding lurus dengan banyaknya jumlah pelaku usaha makanan di bidang jamur. Dalam proses pembesaran jamur pada kumbung sangat tergantung pada faktor fisik seperti suhu dan kelembaban. Jamur tiram dapat menghasilkan tubuh buah secara optimum pada suhu dibawah 30°C dan rentang kelembaban udara 80-90%RH (Relative Humidity). Seiring berjalannya waktu para petani ternyata mengalami hambatan dalam melakukan pengendalian suhu dan kelembaban kumbung jamur sehari-hari akibat fluktuasi suhu dan kelembaban sangat tinggi dan ini menyebabkan hasil panen terganggu.

Pengendalian suhu dan kelembaban yang dilakukan dalam penelitian ini bertujuan untuk merancang dan membuat sistem kontrol yang diterapkan ke dalam miniatur kumbung jamur dengan kontrol logika *fuzzy*. Miniatur kumbung jamur berukuran 80 cm^3 . Aktuator yang digunakan berupa kipas dan pompa air. Sensor yang dipakai adalah SHT11, sebagai pengukur suhu dan kelembaban. Masukan dari kontrol logika *fuzzy* berupa *error* dan *delta error* dari *set point* dikurangi suhu di dalam kumbung jamur. Fungsi keanggotaan masukan dari suhu dan kelembaban masing-masing terdiri dari 5 label. Aturan *fuzzy* yang dibuat ada 2, untuk suhu dan kelembaban, masing – masing 25 macam. Fungsi keanggotaan keluaran dari suhu dan kelembaban berupa singular, untuk mempermudah penyusunan algoritma *fuzzy* dan mempercepat proses defuzzifikasi, dan komponen yang dikontrol disini adalah suhu dan kelembaban, sehingga tidak dibutuhkan tingkat presisi yang tinggi. *Set point* dari suhu adalah 22°C dan untuk kelembaban adalah 90%RH.

Dari hasil penelitian, suhu dan kelembaban yang didapatkan setelah proses kontrol memenuhi kebutuhan jamur tiram untuk tumbuh. Didapatkan *error* suhu secara keseluruhan sistem adalah 3.87% dan untuk kelembaban adalah 3.27%. Sistem dapat mencapai target dari *set point* yang telah ditentukan.

Kata Kunci : jamur tiram, kontrol logika fuzzy, suhu, kelembaban, SHT11

SUMMARY

Ade Barlian Tandiono, Department of Electrical Engineering, Faculty of Engineering, University of Brawijaya, January 2016, *Control of temperature and humidity On the cultivation of Oyster Mushrooms by using Fuzzy Logic Control Method*, Supervisor : M. Aziz Muslim and Moch. Rusli.

Cultivation of oyster mushrooms is currently growing rapidly as can be seen by a growing number of mushroom farmers that is directly proportional to the number of businessmen in the field of mushrooms food. Fungus enlargement process is very dependent on the physical factors such as temperature and humidity. Oyster mushrooms can produce body in optimum operating temperature below 30°C and humidity range 80-90%RH. The farmers apparently experiencing difficulties in controlling temperature and humidity because of daily fluctuation of temperature and humidity is very high and this causes a disrupted harvest.

Control of temperature and humidity done in this study aims to design and create a control system which is applied into miniature mushroom cultivate room with fuzzy logic control. The miniature size is 80cm³. Actuators are fan and water pump. SHT11 is used as a sensor for measuring temperature and humidity. The Fuzzy Logic Controller has 2 inputs, i.e error and delta error of temperature and humidity. Since each input has 5 labels, 25 fuzzy rule were developed for this purpose. The membership function of the output temperature and humidity in the form of the singular, to ease fuzzy algorithms and accelerate the process of defuzzification, and also since controlled variable here are the temperature and humidity, no high level precision is required. The set point of the temperature is 22°C and humidity is 90%RH.

As the research results, the temperature and humidity obtained after the control process meets the needs of oyster mushrooms to grow. It brings the overall temperature error is 3.87% and for humidity is 3.27%. The system can achieve the set point.

Keywords: Oyster Mushrooms, fuzzy logic control, temperature, humidity, SHT11

BAB I

PENDAHULUAN

1.1. Latar Belakang

Jamur tiram (*Pleurotus ostreatus*) adalah jamur pangan dari kelompok *Basidiomycota* dan termasuk kelas *Homobasidiomycetes* dengan ciri-ciri umum tubuh buah berwarna putih hingga krem dan tudungnya berbentuk setengah lingkaran mirip cangkang tiram dengan bagian tengah agak cekung. Jamur tiram masih satu kerabat dengan *Pleurotus eryngii* dan sering dikenal dengan sebutan *King Oyster Mushroom*. (Erie Maulana Sy, 2012: 1)

Perkembangan budidaya jamur tiram saat ini mengalami pertumbuhan yang sangat pesat ditandai dengan semakin banyaknya petani pembibit maupun pembesaran jamur yang berbanding lurus dengan banyaknya jumlah pelaku usaha makanan di bidang jamur. Dalam proses pembesaran jamur pada kumbung sangat tergantung pada faktor fisik seperti suhu dan kelembaban. Jamur tiram dapat menghasilkan tubuh buah secara optimum pada suhu dibawah 30°C dan rentang kelembaban udara 80-90%RH (*relative humidity*).

Seiring berjalannya waktu para petani ternyata mengalami hambatan dalam melakukan pengendalian suhu dan kelembaban kumbung jamur sehari-hari. Upaya yang mereka lakukan biasanya adalah dengan manual menggunakan *sprayer* yang disemprotkan pada kumbung jamur saat pagi dan sore setiap harinya. Selain kurang praktis, ada kelemahan lain yang terjadi dengan metode yang seperti itu, yaitu jika kondisi suhu dan kelembaban berfluktuasi sepanjang hari seperti yang biasa terjadi pada pergantian musim saat ini, maka tidak cukup hanya disemprot dengan *sprayer* tiap pagi dan sore hari. Fluktuasi suhu dan kelembaban sangat tinggi dan ini menyebabkan hasil panen mereka terganggu.

Akan menjadi sebuah permasalahan tersendiri apabila para petani harus sepanjang waktu berada di kumbung jamur untuk melakukan penyemprotan. Oleh karena itu pada skripsi ini, akan dibahas mengenai sistem pengendalian suhu dan kelembaban kumbung jamur tiram dengan Kontrol Logika Fuzzy.

1.2. Rumusan Masalah

Berdasarkan uraian pada latar belakang, dapat disusun rumusan masalah sebagai berikut :

- Bagaimana merancang dan membuat sistem pengendalian suhu dan kelembaban kumbung jamur tiram dengan Kontrol Logika *Fuzzy*.

1.3. Batasan Masalah

Dalam perancangan skripsi ini permasalahan dibatasi oleh hal-hal sebagai berikut :

- 1) Jamur yang dibudidayakan hanya jamur tiram.
- 2) Lebih difokuskan pada pengaturan kondisi lingkungan kumbung jamur.
- 3) Pembahasan ditekankan pada penggunaan kontroler logika *fuzzy*.
- 4) Parameter yang digunakan berupa suhu dan kelembaban.
- 5) Variabel yang di fuzzy-kan adalah *error* dan delta *error* suhu dan kelembaban.
- 6) Penelitian dimulai dr proses tumbuh buah sampai panen.
- 7) Suhu yang dikontrol antara 22-27 °C.
- 8) Kelembaban yang dikontrol antara 80 – 90 %RH.

1.4. Tujuan

Mendapatkan hasil budidaya jamur yang optimal dengan menggunakan Kontrol Logika *Fuzzy* dan mempermudah kinerja petani dalam menjaga pertumbuhan jamur.

1.5. Sistematika Pembahasan

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

BAB I Pendahuluan

Membahas latar belakang, rumusan masalah, batasan masalah, tujuan dan sistematika pembahasan.

BAB II Dasar Teori

Membahas teori-teori yang mendukung dalam perancangan dan pembuatan alat.

BAB III Metodologi Penelitian

Membahas metode penelitian dan perencanaan alat.

BAB IV Perancangan Algoritma Kontrol Logika Fuzzy

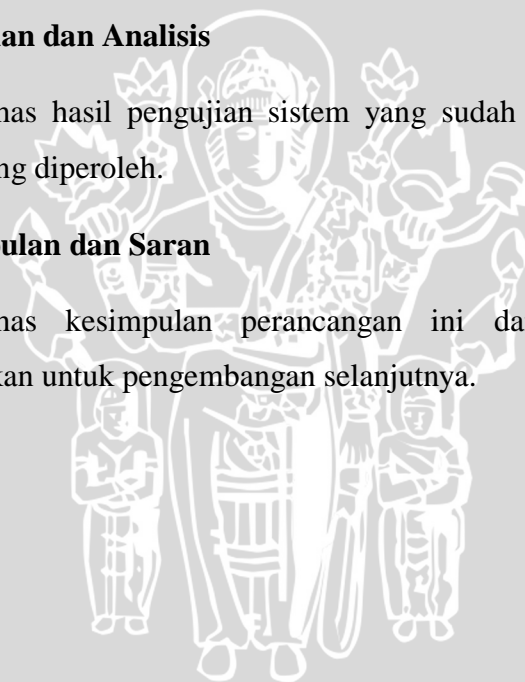
Membahas perancangan kontroler logika *fuzzy* dan menerapkannya kedalam *software*, sehingga sistem dapat bekerja dengan baik.

BAB V Pengujian dan Analisis

Membahas hasil pengujian sistem yang sudah dibuat, serta analisis hasil yang diperoleh.

Bab VI Kesimpulan dan Saran

Membahas kesimpulan perancangan ini dan saran-saran yang diperlukan untuk pengembangan selanjutnya.



BAB II

DASAR TEORI

Pada Bab II ini akan dijelaskan teori penunjang yang digunakan dalam penulisan ini, dengan tujuan untuk memudahkan dalam memahami cara kerja rangkaian maupun dasar-dasar perencanaan alat ini. Teori penunjang yang akan dijelaskan dalam Bab II ini adalah:

- Kontrol Logika Fuzzy
- PWM (*Pulse Width Modulation*)
- Mikrokontroler Arduino Uno
- SHT 11
- Motor DC
- Jamur Tiram

2.1. Kontrol Logika Fuzzy

Fuzzy secara harfiah berarti samar, sedangkan kebalikannya dalam hal ini adalah *Crisp* yang secara harfiah berarti tegas. Dalam kehidupan sehari-hari nilai samar lebih akrab daripada nilai tegas. Temperatur tertentu biasa dinyatakan sebagai panas, agak panas, atau sangat dingin daripada dinyatakan dalam nilai terukur tertentu.

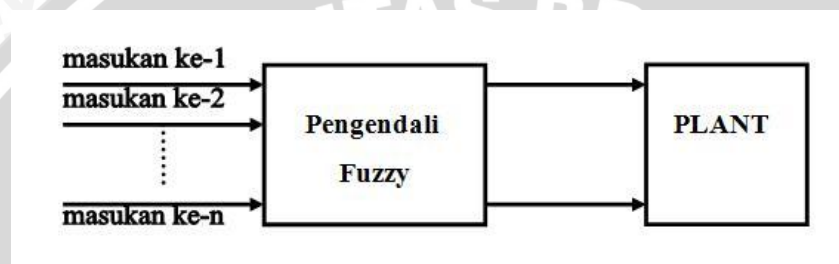
Tahun 1965 L.A. Zadeh memodifikasi teori himpunan yang disebut himpunan kabur (*fuzzy Set*). Himpunan *fuzzy* di dasarkan pada gagasan untuk memperluas jangkauan fungsi karakteristik sehingga fungsi tersebut akan mencakup bilangan real pada interval $[0,1]$. Nilai keanggotaannya menunjukkan bahwa suatu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

Sejak tahun 1982 pengendalian berbasis logika *fuzzy* mengalami perkembangan pesat, terutama dalam hubungannya dengan penyelesaian masalah kendali yang bersifat tak linier,

sulit dimodelkan, berubah karakteristiknya terhadap waktu (*time varying*) dan kompleks (Sivanandam, 2006).

2.1.1. Struktur Dasar Kontrol Logika Fuzzy

Dalam sistem pengendalian dengan logika *fuzzy* melibatkan suatu blok pengendali yang menerima satu atau lebih masukan dan mengumpangkan satu atau lebih keluaran ke plant atau blok lain sebagaimana ditunjukkan dalam Gambar 2.1



Gambar 2.1 Pengendali Fuzzy
Sumber : Coughanowr,1991

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

2.1.2. Fungsi Keanggotaan

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

Suatu himpunan *fuzzy* A dalam semesta pembicaraan U dinyatakan dengan fungsi keanggotaan, μ_A yang harganya berada dalam interval $[0,1]$. Secara matematika hal ini dinyatakan dengan :

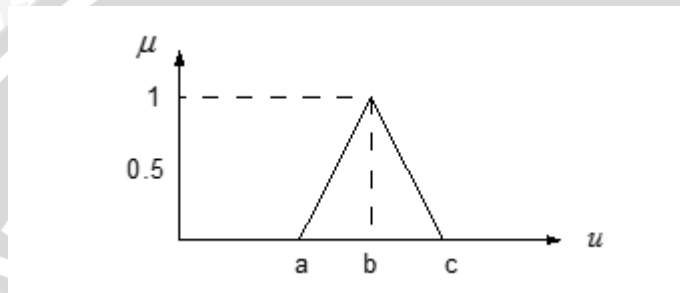
$$\mu_A : U \rightarrow [0,1] \text{ (Kuswadi, 2000)}$$

- Fungsi keanggotaan bentuk Triangular

Definisi fungsi triangular sebagai berikut:

$$T(u; a, b, c) = \begin{cases} 0 & u < a \\ \frac{u-a}{b-a} & a \leq u \leq b \\ \frac{c-u}{c-b} & b \leq u \leq c \\ 0 & u > c \end{cases} \quad (2.1)$$

Fungsi keanggotaan bentuk Tringular ditunjukkan dalam Gambar 2.2.



Gambar 2.2 Fungsi Keanggotaan Bentuk Triangular
Sumber : Yan, 1994

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

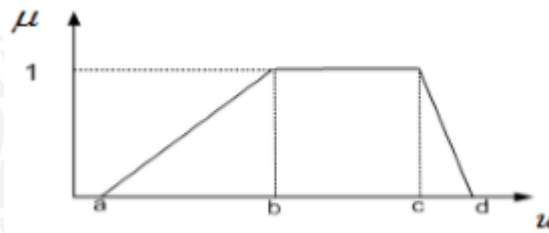
➤ Fungsi keanggotaan bentuk Trapesium

Definisi fungsi trapesium sebagai berikut :

$$T(u; a, b, c, d) = \begin{cases} 0 & u < a \\ \frac{u-a}{b-a} & a \leq u \leq b \\ 1 & b \leq u \leq c \\ \frac{d-u}{d-c} & c \leq u \leq d \\ 0 & d \leq u \end{cases} \quad (2.2)$$



Fungsi keanggotaan bentuk Trapesium ditunjukkan pada Gambar 2.3



Gambar 2.3 Fungsi Keanggotaan Bentuk Trapesium
Sumber : Yan, 1994

2.1.3. Fuzzy Logic Controller

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

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

2.1.4. Fuzzifikasi

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

$$x = \text{fuzzifier}(x_0) \quad (2.3)$$

dengan:

x_0 = nilai *crisp* variabel masukan

x = himpunan *fuzzy* variabel yang terdefinisi

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

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

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

2.1.5. Kaidah Aturan Fuzzy (Fuzzy Rule)

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

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

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

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

·
·
·

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

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

2.1.6. Metode Inferensi MAX-MIN

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

Pada metode Max–Min aturan operasi minimum Mamdani digunakan untuk implikasi *fuzzy*. Persamaan aturan minimum adalah

$$\mu_{c_i} = \bigcup_1^n \alpha_i \wedge \mu_{c_i} \quad (2.4)$$

dengan $\alpha_i = \mu_{A_i}(x_0) \wedge \mu_{B_i}(y_0)$

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

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

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

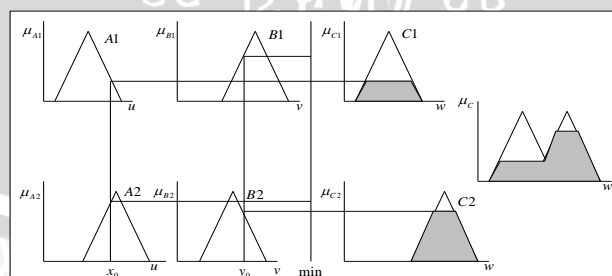
Pada metode penalaran MAX-MIN fungsi keanggotaan konsekuen dinyatakan dengan

$$\mu_{c_1}(W) = \mu_{c_1} \vee \mu_{c_2} = [\alpha_1 \wedge \mu_{c_1}(w)] \vee [\alpha_2 \wedge \mu_{c_2}(w)] \quad (2.5)$$

$$\text{dimana } \alpha_1 = \mu_{A_1}(x_0) \wedge \mu_{B_1}(y_0) \quad (2.6)$$

$$\alpha_2 = \mu_{A_2}(x_0) \wedge \mu_{B_2}(y_0) \quad (2.7)$$

Lebih jelas metode ini dideskripsikan dalam Gambar 2.4



Gambar 2.4 Inferensi Fuzzy dengan Metode MAX-MIN

Sumber : Yan, 1994

2.1.7. Metode Defuzzifikasi *Center Of Gravity* (COG)

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

$$y_0 = defuzzifier(y) \tag{2.8}$$

dengan:

- y : aksi kontrol *fuzzy*
- y₀ : aksi kontrol *crisp*
- defuzzifier* : operator defuzzifikasi

Metode *Center Of Gravity* didefinisikan sebagai berikut:

$$U = \frac{\sum_{i=1}^n w_i u_i}{\sum_{i=1}^n w_i} \tag{2.9}$$

dengan:

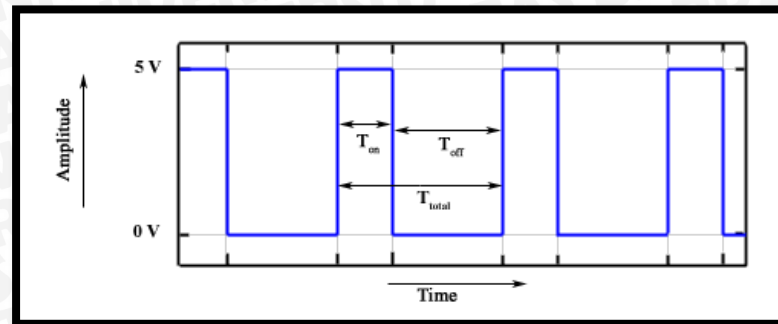
- U = Keluaran
- w_i = Bobot nilai benar w_i
- u_i = Nilai linguistik pada fungsi keanggotaan keluaran
- n = Banyak derajat keanggotaan

2.2. PWM (*Pulse Width Modulation*)

PWM (*Pulse Width Modulation*) digunakan untuk mengatur kecepatan dari motor DC. Kecepatan motor DC tergantung pada besarnya *duty cycle* yang diberikan pada motor DC tersebut.

Pada sinyal PWM, frekuensi sinyal konstan sedangkan *duty cycle* bervariasi dari 0% - 100%. Dengan mengatur *duty cycle* akan diperoleh keluaran yang diinginkan. Sinyal PWM (*Pulse Width Modulation*) secara umum dapat dilihat dalam gambar 2.5 berikut:





Gambar 2.5 Sinyal PWM Secara Umum
Sumber: electronics-scheme.com

$$Duty\ cycle = \frac{T_{on}}{T} \times 100\% \dots (\%) \quad (2.10)$$

Dengan:

T_{on} = Periode logika tinggi

T = Periode keseluruhan

$$V_{dc} = Duty\ cycle \times V_{cc} \dots (V) \quad (2.11)$$

Sedangkan frekuensi sinyal dapat ditentukan dengan rumus berikut:

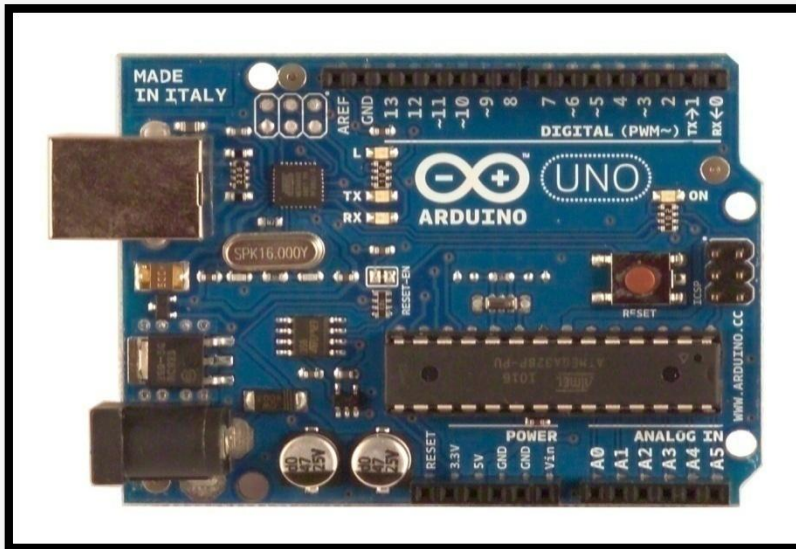
$$f_{On} = \frac{f_{clk}/O}{N.256} \dots (Hz) \quad (2.12)$$

2.3. Arduino Uno

Arduino Uno adalah board mikrokontroler berbasis ATmega328. Memiliki 14 pin *input* dari *output* digital di mana 6 pin *input* tersebut dapat digunakan sebagai *output* PWM dan 6 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 Uno ke komputer dengan menggunakan kabel USB atau listrik dengan AC yang ke adaptor-DC atau baterai untuk menjalankannya.

Uno berbeda dengan semua *board* sebelumnya dalam hal koneksi USB-to-serial yaitu menggunakan fitur Atmega8U2 yang diprogram sebagai konverter USB-to-serial

berbeda dengan *board* sebelumnya yang menggunakan chip FTDI *driver* USB-to-serial. Tampak depan Arduino Uno dapat dilihat pada Gambar 2.6.



Gambar 2.6 Tampak depan Arduino Uno

Nama “Uno” berarti *satu* dalam bahasa Italia, untuk menandai peluncuran Arduino 1.0. Uno dan versi 1.0 akan menjadi versi referensi dari Arduino. Uno adalah yang terbaru dalam serangkaian *board* USB Arduino, dan sebagai model referensi untuk platform Arduino, untuk perbandingan dengan versi sebelumnya,

2.3.1. Daya

Arduino Uno dapat diaktifkan melalui koneksi USB atau dengan catu daya eksternal (otomatis). Eksternal (non-USB) daya dapat berasal baik dari AC-ke adaptor-DC atau baterai. Adaptor ini dapat dihubungkan dengan menancapkan plug jack pusat-positif ukuran 2.1mm konektor POWER. Ujung kepala dari baterai dapat dimasukkan kedalam Gnd dan Vin pin *header* dari konektor POWER. Kisaran kebutuhan daya yang disarankan untuk board Uno adalah 7 sampai dengan 12 volt, jika diberi daya kurang dari 7 volt kemungkinan pin 5v Uno dapat beroperasi tetapi tidak stabil kemudian jika diberi daya lebih dari 12V, regulator tegangan bisa panas dan dapat merusak board Uno.

Pin listrik adalah sebagai berikut:

VIN. Tegangan masukan kepada board Arduino ketika itu menggunakan sumber daya eksternal (sebagai pengganti dari 5 volt koneksi USB atau sumber daya lainnya).5V. Catu daya digunakan untuk daya mikrokontroler dan komponen lainnya. Sebuah pasokan 3,3 volt dihasilkan oleh regulator *on-board*.

GND. *Ground* pin.

2.3.2. Memori

ATmega328 memiliki 32 KB (dengan 0,5 KB digunakan untuk bootloader), 2 KB dari SRAM dan 1 KB EEPROM.

2.3.3. Input dan Output

Masing-masing dari 14 pin digital di Uno dapat digunakan sebagai input atau output, dengan menggunakan fungsi *pinMode* (), *digitalWrite* (), dan *digitalRead* (), beroperasi dengan daya 5 volt. Setiap pin dapat memberikan atau menerima maksimum 40 mA dan memiliki internal *pull-up* resistor (secara *default* terputus) dari 20-50 kOhms. Selain itu, beberapa pin memiliki fungsi khusus:

Serial: 0 (RX) dan 1 (TX). Digunakan untuk menerima (RX) dan mengirimkan (TX) TTL data serial. Pin ini dihubungkan ke pin yang berkaitan dengan chip Serial ATmega8U2 USB-to-TTL.

Eksternal menyela: 2 dan 3. Pin ini dapat dikonfigurasi untuk memicu *interrupt* pada nilai yang rendah, dengan batasan tepi naik atau turun, atau perubahan nilai.

PWM: 3, 5, 6, 9, 10, dan 11. Menyediakan *output* PWM 8-bit dengan fungsi *analogWrite* ().

SPI: 10 (SS), 11 (Mosi), 12 (MISO), 13 (SCK). Pin ini mendukung komunikasi SPI menggunakan *SPI library*.

LED: 13. Ada *built-in* LED terhubung ke pin digital 13. Ketika pin bernilai nilai HIGH, LED on, ketika pin bernilai LOW, LED off.

Uno memiliki 6 masukan analog, berlabel A0 sampai dengan A5, yang masing-masing menyediakan 10 bit dengan resolusi (yaitu 1024 nilai yang berbeda). Selain itu, beberapa pin memiliki fungsi khusus:

I2C: A4 (SDA) dan A5 (SCL). Dukungan I2C (TWI) komunikasi menggunakan perpustakaan *Wire*.

Aref. Tegangan referensi (0 sampai 5V saja) untuk *input* analog. Digunakan dengan fungsi *analogReference ()*.

Reset. Bawa baris ini LOW untuk me-reset mikrokontroler.

2.3.4. Komunikasi

Arduino Uno memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, Arduino lain, atau mikrokontroler lainnya. ATmega328 menyediakan UART TTL (5V) untuk komunikasi serial, yang tersedia di pin digital 0 (RX) dan 1 (TX). Sebuah ATmega8U2 sebagai saluran komunikasi serial melalui USB dan sebagai port virtual com untuk perangkat lunak pada komputer. *Firmware* '8 U2 menggunakan *driver* USB standar COM, dan tidak ada *driver* eksternal yang diperlukan. Namun, pada Windows diperlukan, sebuah file inf. Perangkat lunak Arduino terdapat monitor serial yang memungkinkan digunakan memonitor data tekstual sederhana yang akan dikirim ke atau dari board Arduino. LED RX dan TX di papan tulis akan berkedip ketika data sedang dikirim melalui chip USB-to-serial dengan koneksi USB ke komputer (tetapi tidak untuk komunikasi serial pada pin 0 dan 1).

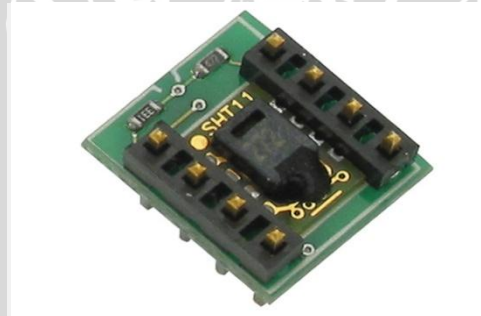
Sebuah *Software Serial library* memungkinkan untuk berkomunikasi secara serial pada salah satu pin digital pada *board* Uno's. ATmega328 juga mendukung I2C (TWI) dan komunikasi SPI. Perangkat lunak Arduino termasuk perpustakaan *Kawat* untuk menyederhanakan penggunaan bus I2C.

2.4. SHT 11

2.4.1. Pengertian

Sensor suhu dan kelembaban adalah peralatan yang digunakan untuk merubah besaran panas menjadi besaran listrik yang nantinya dapat dianalisis hasil besarnya dari suhu yang dikeluarkan oleh suatu tempat yang dipantau keadaan suhunya oleh sensor suhu itu.

SHT11 adalah sensor digital untuk temperatur sekaligus kelembapan pertama didunia yang diklaim oleh pabrik pembuatnya yaitu Sensirion Corp yang mempunyai kisaran pengukuran dari 0-100% RH dan akurasi RH absolut +/- 3% RH. Sedangkan akurasi pengukuran temperatur +/- 0.4°C pada suhu 25 °C. Modul sensor ini sudah memiliki keluaran digital dan sudah terkalibrasi, jadi pengguna tidak perlu lagi melakukan konversi A/D ataupun kalibrasi data sensor. Antarmuka modul ini adalah serial 2-wire (bukan I2C) sehingga sangat menghemat jalur I/O kontroler. Bentuk dari sensor SHT11 dapat dilihat pada Gambar 2.7.



Gambar 2.7 Sensirion Temperature/Humidity Sensor - SHT11

Sumber: www.adafruit.com

2.4.2. Prinsip Kerja Sensor SHT 11

SHT11 adalah sebuah single chip sensor suhu dan kelembaban relatif dengan multi modul sensor yang outputnya telah dikalibrasi secara digital. Dibagian dalamnya terdapat kapasitas polimer sebagai elemen untuk sensor kelembaban relatif dan sebuah pita regangan yang digunakan sebagai sensor temperatur. Output kedua sensor digabungkan

dan dihubungkan pada ADC 14 bit dan sebuah interface serial pada satu chip yang sama. Sensor ini menghasilkan sinyal keluaran yang baik dengan waktu respon yang cepat. SHT11 ini dikalibrasi pada ruangan dengan kelembaban yang teliti menggunakan hygrometer sebagai referensinya. Koefisien kalibrasi telah diprogramkan kedalam OTP memory. Koefisien tersebut akan digunakan untuk mengkalibrasi keluaran dari sensor selama proses pengukuran. Sistem sensor yang digunakan untuk mengukur suhu dan kelembaban adalah SHT11 dengan sumber tegangan 5 Volt, sistem sensor ini mempunyai 1 jalur data yang digunakan untuk perintah pengalamatan dan pembacaan data. Pengambilan data untuk masing-masing pengukuran dilakukan dengan memberikan perintah pengalamatan oleh mikrokontroler komunikasi yang digunakan menggunakan antarmuka *two-wire* serial.

2.5. Motor DC

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

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

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

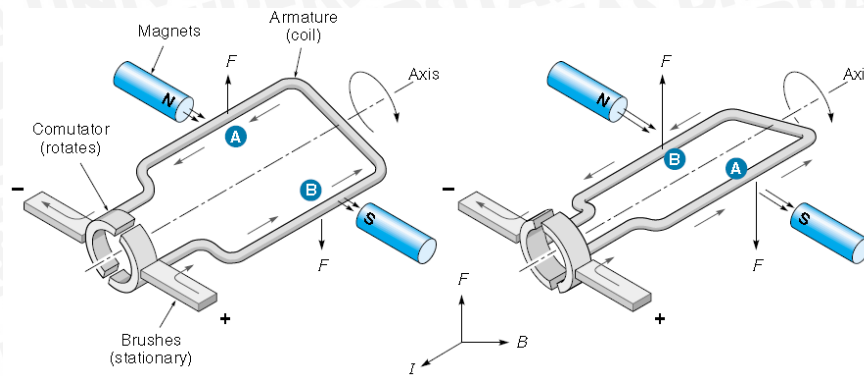
Dimana :

B = kerapatan fluks magnet (weber)

L = panjang konduktor (meter)

I = arus listrik (ampere)

(Soemarwanto, 1999)



Gambar 2.8 Motor DC
Sumber : Kilian, 1996

Gambar 2.8 adalah ilustrasi cara kerja motor DC yang mempunyai satu lilit kawat a-b berada di dalam medan magnet. Lilitan ini dapat berputar dengan bebas, lilitan ini biasa disebut dengan jangkar (armature).

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

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

Bagian *commutator* berfungsi sebagai penyearah mekanik. Fluksi magnet yang ditimbulkan magnet permanen disebut medan magnet motor. Dalam gambar 2.8 arah fluks magnetic adalah dari kiri ke kanan. Adapun gaya yang bekerja pada penghantar b adalah ke atas, sementara gaya yang bekerja pada penghantar a adalah ke bawah. Gaya-gaya yang bekerjasama kuatnya, sehingga terdapat kopel yang bekerja pada kawat sehingga lilitan jangkar dapat berputar. Setelah berputar 180° arah arus berbalik, pada saat itu penghantar a dan b bertukar tempat. Akibatnya arah gerak putaran tidak berubah.

2.6. Jamur Tiram

Salah satu jamur yang menjadi konsumsi favorit adalah jamur tiram. Jamur tiram memiliki ciri-ciri umum tubuh buah berwarna putih hingga krem dan tudungnya berbentuk setengah lingkaran mirip cangkang tiram dengan bagian tengah agak cekung. Jamur tiram sering dikenal dengan sebutan King Oyster Mushroom.

Tubuh buah jamur tiram memiliki tangkai yang tumbuh menyamping (bahasa Latin: Pleurotus) dan bentuknya seperti tiram (*ostreatus*) sehingga jamur tiram mempunyai nama binomial *Pleurotus ostreatus*. Bagian tudung dari jamur tersebut berubah warna dari hitam, abu-abu, coklat, hingga putih, dengan permukaan yang hampir licin, diameter 5-20 cm yang bertepi tudung mulus sedikit berlekuk. Selain itu, jamur tiram juga memiliki spora berbentuk batang berukuran $8-11 \times 3-4 \mu\text{m}$ serta miselia berwarna putih yang bisa tumbuh dengan cepat.

Secara alami, jamur tiram *Pleurotus* ditemukan di hutan dibawah pohon berdaun lebar atau di bawah tanaman berkayu. Jamur tiram tidak memerlukan cahaya matahari yang banyak, di tempat terlindung misellium jamur akan tumbuh lebih cepat daripada di tempat yang terang dengan cahaya matahari berlimpah. Pertumbuhan misellium akan tumbuh dengan cepat dalam keadaan gelap/tanpa sinar. Pada masa pertumbuhan misellium, jamur tiram sebaiknya ditempatkan dalam ruangan yang gelap, tetapi pada masa pertumbuhan badan buah memerlukan adanya rangsangan sinar. Pada tempat yang sama sekali tidak ada cahaya badan buah tidak dapat tumbuh, oleh karena itu pada masa terbentuknya badan buah pada permukaan media harus mulai mendapat sinar dengan intensitas penyinaran 60 - 70 %.

Dalam menggunakan media pertumbuhan, jerami yang baik untuk dibuat sebagai bahan media tanam adalah dari jenis jerami yang keras sebab jerami yang keras banyak mengandung selulosa yang merupakan bahan yang diperlukan oleh jamur dalam jumlah banyak disamping itu jerami yang keras membuat media tanaman tidak cepat habis. Hal yang perlu diperhatikan dalam pemilihan jerami sebagai bahan baku media tanam adalah dalam hal kebersihan dan kekeringan, selain itu jerami yang digunakan tidak busuk dan tidak ditumbuhi jamur jenis lain. Media yang terbuat dari campuran bahan-bahan tersebut

perlu diatur kadar airnya. Kadar air diatur 60 - 65 % dengan menambah air bersih agar misellia jamur dapat tumbuh dan menyerap makanan dari media tanam dengan baik.

Selain jerami, media lain yang dapat digunakan seperti media serbuk gergaji yang mengandung selulosa, lignin, pentosan, zat ekstraktif, abu, jerami padi, media limbah kapas, alang-alang, daun pisang, tongkol jagung, klobot jagung, gabah padi, dan lain sebagainya. Tetapi, tetap saja pertumbuhan yang paling baik ada di media serbuk gergaji dan merang. Penyebabnya adalah karena jumlah lignoselulosa, lignin, dan serat pada serbuk gergaji dan merang memang lebih tinggi. Sebagai contohnya dalam pembuatan media jerami padi, bahan-bahan yang digunakan adalah 15-20% jerami padi, 2.5% bekatul kaya karbohidrat, karbon, dan vitamin B kompleks yang bisa mempercepat pertumbuhan dan mendorong perkembangan tubuh buah jamur, 1-1.5% kalsium karbonat atau kapur menetralkan media sehingga dapat ditumbuhi oleh jamur (pH 6 – 7). Selain itu, kapur juga mengandung kalsium sebagai penguat batang / akar jamur agar tidak mudah rontok. 0.5% gips dapat memperkokoh struktur suatu bahan campuran, dan terakhir 0.25% pupuk TS sebagai nutrisi.

Pada budidaya jamur tiram suhu udara memegang peranan yang penting untuk mendapatkan pertumbuhan badan buah yang optimal. Pada umumnya suhu yang optimal untuk pertumbuhan jamur tiram, dibedakan dalam dua fase yaitu fase inkubasi yang memerlukan suhu udara berkisar antara 22 - 28 °C dengan kelembapan 60 - 70 %RH dan fase pembentukan tubuh buah memerlukan suhu udara antara 16 - 22 °C dengan kelembapan lingkungan cukup tinggi 95 – 98 %RH. (Erie Maulana Sy, 2012: 1)

Kondisi di atas lebih mudah dicapai di daerah dataran tinggi sekitar 700-800 m dpl. Kemungkinan budidaya jamur di dataran rendah tidaklah mustahil asalkan iklim ruang penyimpanan dapat diatur dan disesuaikan dengan keperluan jamur.

BAB III

METODOLOGI PENELITIAN

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

3.1. Penentuan Spesifikasi Alat

Spesifikasi alat secara umum ditetapkan terlebih dahulu sebagai acuan dalam perancangan selanjutnya. Spesifikasi alat yang direncanakan adalah sebagai berikut :

- Menggunakan satu buah motor DC dan pompa mobil isuzu
- Menggunakan kipas dan *sprayer* untuk mengatur suhu dan kelembaban.
- Sensor suhu dan kelembaban yang digunakan adalah SHT 11.
- Mikrokontroler yang digunakan Arduino Uno Rev3.
- Kontroler yang digunakan adalah kontroler logika fuzzy.

3.2. Perancangan Sistem

Perancangan sistem dilakukan sebagai langkah awal sebelum terbentuknya suatu sistem beserta rangkaian elektronik pendukungnya, hal ini dimaksudkan agar sistem dapat berjalan sesuai dengan deskripsi awal yang telah direncanakan.

Perancangan sistem yang dilakukan meliputi:

1. Penentuan sensor yang akan digunakan untuk mendeteksi suhu dan kelembaban.
2. Penentuan motor DC yang akan digunakan untuk menyalakan kipas.
3. Penentuan pompa air yang akan digunakan untuk *sprayer*.
4. Penentuan spesifikasi sistem yang akan dibuat, meliputi :
 - a. Penentuan deskripsi kerja sistem secara keseluruhan.
 - b. Kontroler logika fuzzy dan rangkaian elektronik pendukung.
 - c. Perancangan program pada Mikrokontroler arduino sebagai pusat pengontrol dengan kontrol logika fuzzy.

3.3. Realisasi Pembuatan Sistem

3.3.1. Perangkat Keras

Komponen utama perangkat keras adalah sebuah miniatur kumbung jamur berukuran 80cm x 80cm x 80cm, motor DC, pompa mobil isuzu, sensor SHT 11 untuk mendeteksi suhu dan kelembaban dalam kumbung, kipas dan *sprayer* sebagai aktuator, dan mikrokontroler arduino uno rev3.

3.3.2. Perangkat Lunak

Pembuatan perangkat lunak (*software*) dilakukan dengan pembuatan *flowchart* terlebih dahulu, penentuan fungsi keanggotaan masukan, aturan fuzzy, fungsi keanggotaan keluaran, kemudian penentuan metode inferensi dan defuzzifikasi, kemudian dilakukan perancangan program pada *software* Arduino Uno Rev3 sebagai pusat pengontrol dengan kontrol logika fuzzy.

3.4. Pengujian Alat

Setelah semua komponen pada alat sudah terhubung sesuai dengan diagram blok sistem yang telah dirancang dan perangkat lunak untuk mendukung sistem telah dibuat, maka diadakan pengujian dan analisa alat. Metode pengujian alat adalah sebagai berikut:

1. Menguji pada tiap-tiap blok rangkaian.
2. Menguji sensor yang digunakan.
3. Menggabungkan beberapa blok rangkaian menjadi sebuah sistem.
4. Mengadakan pengujian sistem secara keseluruhan.
5. Mengevaluasi hasil pengujian sistem secara keseluruhan

3.5. Pengambilan Kesimpulan dan Saran

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

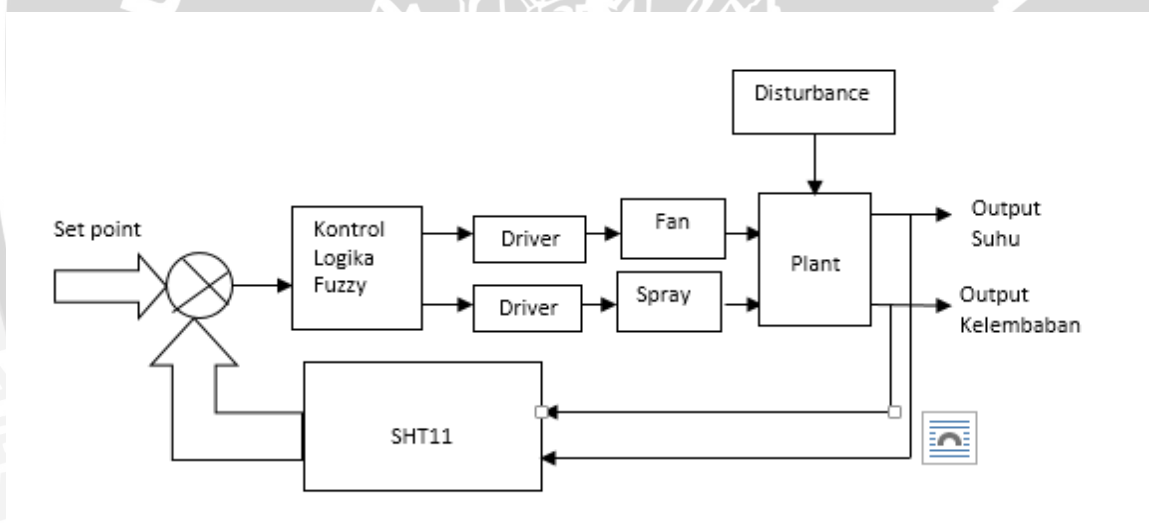
BAB IV

PERANCANGAN & PEMBUATAN ALAT

Pada bab ini akan dijelaskan mengenai perancangan dan pembuatan kumbung jamur mulai dari diagram blok sistem, desain mekanik, perancangan perangkat keras, dan perancangan perangkat lunak. Perancangan dan pembuatan dilakukan secara bertahap dan sistematis, sehingga nantinya akan memudahkan dalam analisis sistem.

4.1. Diagram Blok Sistem

Pada perancangan alat diperlukan perancangan blok diagram sistem yang dapat menjelaskan sistem secara garis besar dan diharapkan alat dapat bekerja sesuai dengan rencana. Blok diagram tersebut dapat dilihat dari Gambar 4.1 berikut.



Gambar 4.1 Blok Diagram Sistem
(Perancangan)

Keterangan dari blok diagram di atas adalah sebagai berikut :

1. Sensor yang digunakan adalah SHT11 yang berfungsi mendeteksi suhu dan kelembaban pada kumbung jamur dan akan memberikan masukan berupa pulsa ke Arduino.
2. Pusat pengendalian sistem menggunakan Arduino Uno.

3. Motor yang digunakan adalah motor DC yang berfungsi sebagai aktuator kipas. *Driver* motor *sprayer* menggunakan pompa mobil isuzu 12 volt.
4. *Disturbance* atau gangguan pada sistem adalah suhu dan kelembaban di luar kumbung jamur.

4.2. Prinsip Kerja Alat

Cara kerja alat adalah sebagai berikut :

1. Catu daya yang digunakan sebesar 12 volt untuk motor DC dan 5 volt untuk sensor *SHT11*.
2. Sensor *SHT11* sebagai sensor suhu dan kelembaban dalam kumbung jamur. Sensor *SHT11* akan memberikan sinyal kepada arduino untuk mulai menyalakan kipas dan *sprayer*.
3. Selanjutnya sensor *SHT11* akan mensensor suhu dan kelembaban didalam kumbung jamur, lalu digunakan sebagai *input* untuk arduino yang selanjutnya akan diolah oleh kontroler yang ada pada arduino, kemudian arduino akan memberikan sinyal kepada *driver* untuk menyalakan kipas dan *sprayer*.
4. Apabila suhu dan kelembaban sudah mencapai setpoint, maka kontroler akan memberikan sinyal kepada *driver* untuk berhenti.

4.3. Perancangan Perangkat Keras

4.3.1. Spesifikasi Alat

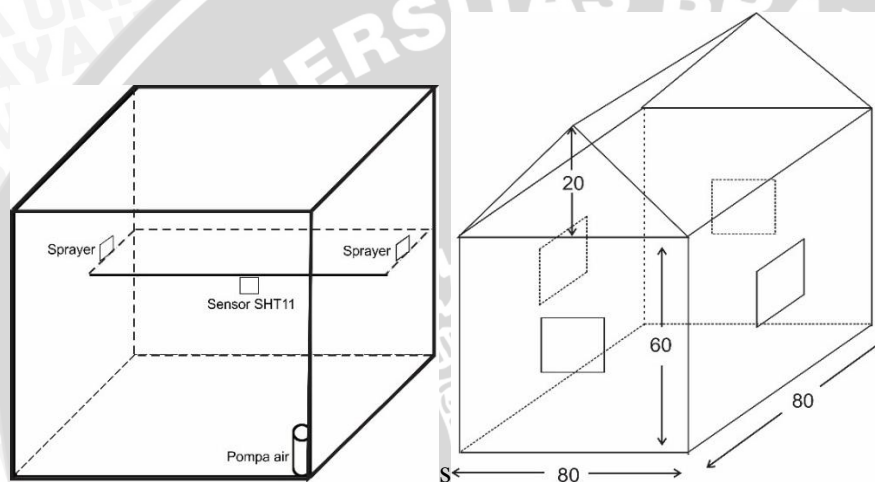
Spesifikasi alat yang dirancang adalah sebagai berikut:

1. Miniatur kumbung jamur dengan ukuran:

Panjang	: 80 cm
Lebar	: 80 cm
Tinggi	: 80 cm

2. Penggerak kipas dan *sprayer* berupa motor DC dengan catu daya 12 volt dan pompa mobil isuzu 12 volt.
3. Sensor yang digunakan untuk mendeteksi suhu dan kelembaban adalah sensor *SHT11*.
4. Kontroler yang digunakan adalah kontrol logika *fuzzy*.
5. Menggunakan satu buah mikrokontroler Arduino *Uno Rev3*.
6. Software yang digunakan sebagai pemrograman yaitu Arduino ERW 1.0.5.

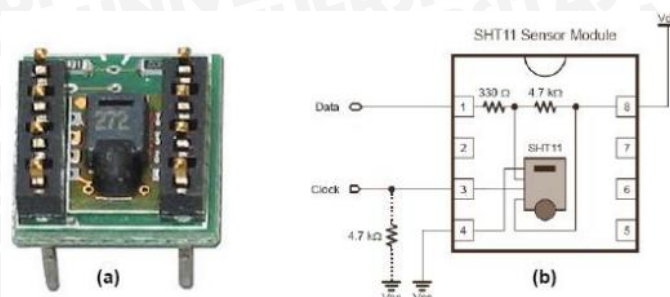
Perancangan perangkat keras kumbung jamur dapat dilihat pada Gambar 4.2



Gambar 4.2 Mekanik Miniatur Kumbung Jamur (Perancangan)

4.3.2 Sensor *SHT11*

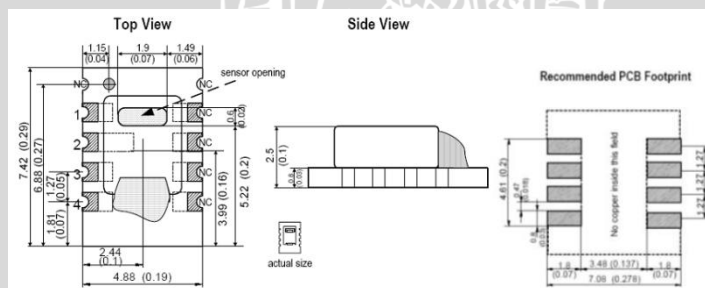
Sensor *SHT11* digunakan sebagai pengukur suhu dan kelembaban di dalam miniatur kumbung jamur. *SHT11* adalah sebuah *single chip* multisensor untuk sensor suhu dan kelembaban ruang yang telah terkalibrasi sempurna sehingga bentuk keluaran sudah dalam bentuk digital. Sensor ini mempunyai akurasi kelembaban kisaran pengukuran dari 0-100% RH dan akurasi RH absolut $\pm 3\%$ RH. Sedangkan akurasi pengukuran suhu $\pm 0.4^{\circ}\text{C}$ pada suhu 25°C . Modul sensor ini sudah memiliki keluaran digital dan sudah terkalibrasi, jadi pengguna tidak perlu lagi melakukan konversi A/D ataupun kalibrasi data sensor. Antarmuka modul ini adalah *serial 2-wire* (bukan I2C) sehingga sangat menghemat jalur I/O kontroler.



Gambar 4.3 Bentuk fisik SHT yang berupa : (a) Modul (b) Skematik [6]

SHT 11 merupakan suatu sensor untuk mengetahui suhu dan kelembaban di udara. Biasanya sensor ini sudah dalam bentuk modul dengan antarmuka *Two-Wire Serial Interface*. Dengan spesifikasi sebagai berikut

- Range suhu : -40°C hingga $+123,8^{\circ}\text{C}$
- Akurasi suhu : $\pm 0,5^{\circ}\text{C}$ pada 25°C
- Range kelembaban : 0 hingga 100% RH
- Akurasi RH absolut : $\pm 3,5\%$ RH
- Faktor bentuk : 8 pin DIP – 0,6"
- Konsumsi daya rendah (tipikal $30\ \mu\text{W}$)
- Tegangan *supply* +5 VDC



Gambar 4.4SHT11 drawing and footprint dimensions in mm (inch) [6]

4.3.3. Rangkaian *Driver* Motor DC

Modul pengendali motor DC yang digunakan adalah modul EMS 30A *H-Bridge*. Secara garis besar, fungsi modul pengendali motor ini adalah untuk mengendalikan kipas dan *sprayer* sesuai instruksi kendali dari Arduino Uno pengguna.

Rangkaian *driver* ini sanggup bekerja dengan tegangan maksimal 40 volt, serta kapasitas arus maksimum yang dapat dilewatkan pada modul ini sebesar 30 ampere. Gambar modul rangkaian driver EMS 30A *H-Bridge* dapat dilihat pada Gambar 4.5



Gambar 4.5 Modul Rangkaian *Driver* EMS 30A *H-Bridge* (www.Arduino.cc)

4.3.4. Liquid Crystal Display (LCD)

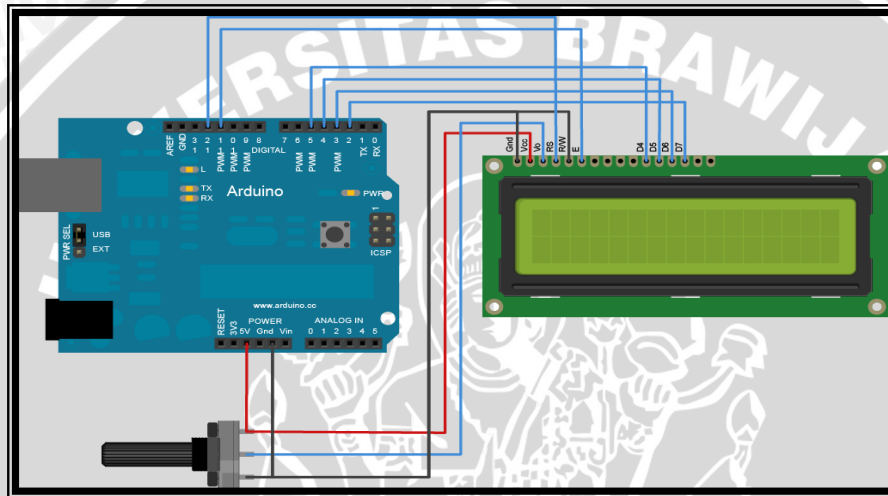
LCD atau *Liquid Crystal Display* yang digunakan dalam perancangan kali ini adalah Hitachi HD44780 dan di dalam sistem ini berfungsi sebagai penampil atau penunjuk suhu dan kelembaban yang ditunjukkan. Digunakannya LCD dalam perancangan ini diperlukan karena apabila hasil atau output suhu dan kelembaban ditunjukkan pada serial monitor yang ada pada Arduino Uno, untuk pengambilan data sebagai bahan analisis perancangan akan menjadi lebih sulit.

LCD Hitachi dapat dikontrol dalam dua mode : 4-bit atau 8-bit. Mode 4-bit membutuhkan tujuh I/O pin Arduino Uno, sedangkan mode 8-bit membutuhkan sebelas I/O pin Arduino Uno. Untuk rangkaian LCD, diperlukan beberapa hardware sebagai penunjang agar LCD dapat berfungsi dengan baik, yaitu :

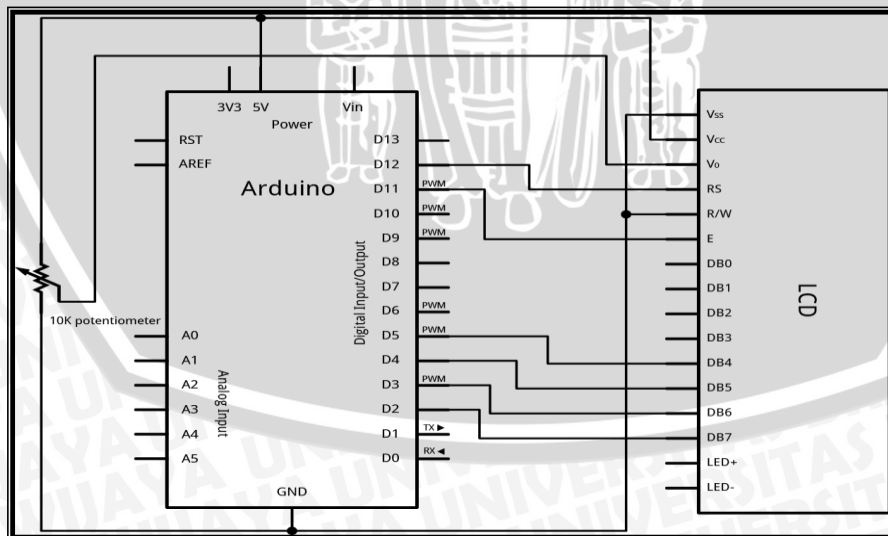
- Arduino Uno
- LCD screen (yang kompatibel dengan Hitachi HD44780)

- Pin header untuk LCD *display*
- 10k potensiometer
- *Project board*
- Kabel *male to female*

Dengan pin header disolder terlebih dulu pada LCD, kemudian dengan menggunakan kabel *male to female* dihubungkan ke Arduino Uno seperti yang ditunjukkan pada Gambar 4.6.



Gambar 4.6 LCD ke Arduino Uno
www.arduino.cc/en/Tutorial/LiquidCrystal

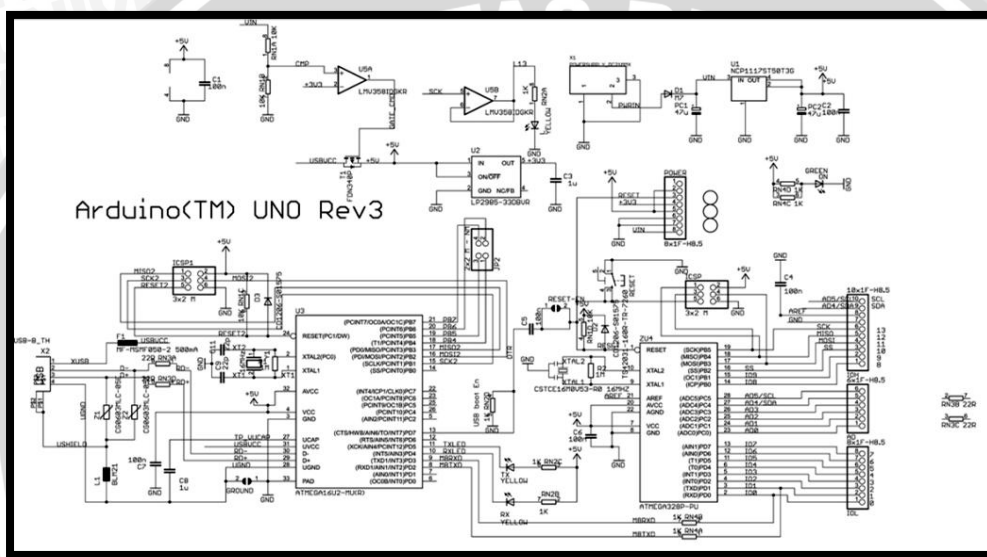


Gambar 4.7 Rangkaian Wiring LCD ke Arduino Uno
www.arduino.cc/en/Tutorial/LiquidCrystal

4.3.5. Modul Arduino Uno Rev3

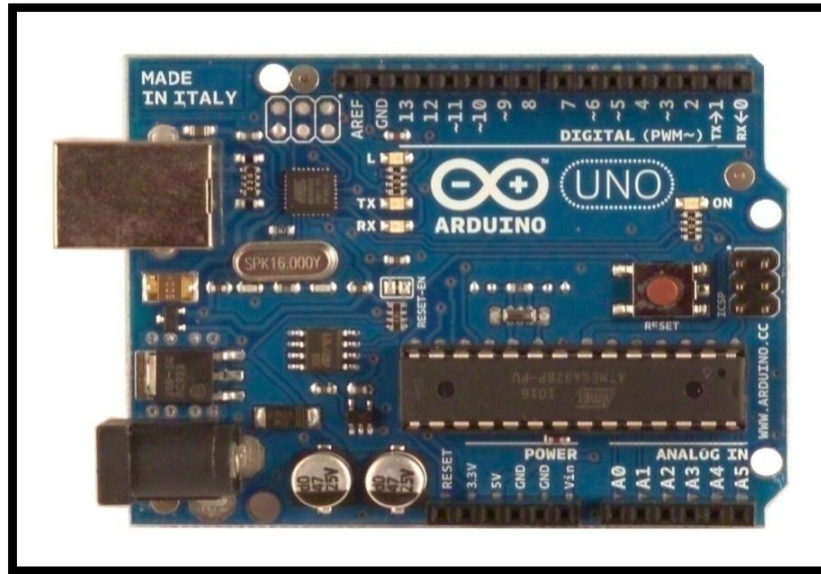
Modul arduino yang digunakan adalah jenis Arduino Uno Rev3. Arduino tersebut digunakan sebagai kontroler, dan pemberi sinyal PWM. Arduino mampu menerima tegangan masuk yaitu antara 0-5V. Untuk mengontrol plant, Arduino perlu diprogram menggunakan Arduino ERW 1.0.5. Sedangkan output yang dikeluarkan oleh arduino dapat berupa sinyal PWM.

Konfigurasi kaki I/O dari Arduino Uno Rev3 ditunjukkan dalam Gambar 4.8



Gambar 4.8 Desain Sistem Arduino Uno Rev3
(Arduino Uno Rev3 Datasheet)

Arduino Uno adalah board mikrokontroler berbasis ATmega328. Memiliki 14 pin input dari output digital dimana 6 pin input tersebut dapat digunakan sebagai output PWM dan 6 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 Uno ke komputer dengan menggunakan kabel USB atau listrik dengan AC yang ke adaptor-DC atau baterai untuk menjalankannya. Modul arduino uno ditunjukkan pada Gambar 4.9



Gambar 4.9 Modul Arduino Uno
(www.Arduino.cc)

Fungsi dari masing-masing pin Arduino Uno dapat dilihat di Tabel 4.1.

Tabel 4.1 Fungsi Pin Arduino Uno Rev3

No	Pin	Fungsi
1	2	Jalur masukan data dari sensor
2	3	LCD pin D6
3	4	LCD pin D5
4	5	LCD pin D4
5	6	Jalur masukan PWM
6	7	Pin input untuk MOUT1
No	Pin	Fungsi
7	8	Pin input untuk MOUT2
8	9	Pin enable untuk MOUT1

9	10	LCD pin D7
10	11	LCD pin Enable
11	12	LCD pin RS
12	5V	Jalur keluaran 5V
13	GND	Jalur keluaran ground

4.4. Perancangan Sistem Kontrol LogikaFuzzy

4.4.1. Variabel Masukan dan Keluaran

Sistem kontrol logika *fuzzy* yang dikembangkan dalam penelitian ini mempunyai dua *crisp input* yaitu *Error*posisi dan delta *Error*posisi serta satu *crispoutput* yaitu perubahan tegangan. *Error* dan $\Delta Error$, didefinisikan dengan perumusan sebagai berikut:

$$Error(t) = SP - PV(t) \quad (4.1)$$

Dimana,

SP = *Set point* (Nilai yang diinginkan)

$PV(t)$ = *Present Value* pada waktut (Nilai aktual)

$$\Delta Error(t) = Error(t) - Error(t-1) \quad (4.2)$$

Dimana,

$Error(t)$ = *Error* pada waktut

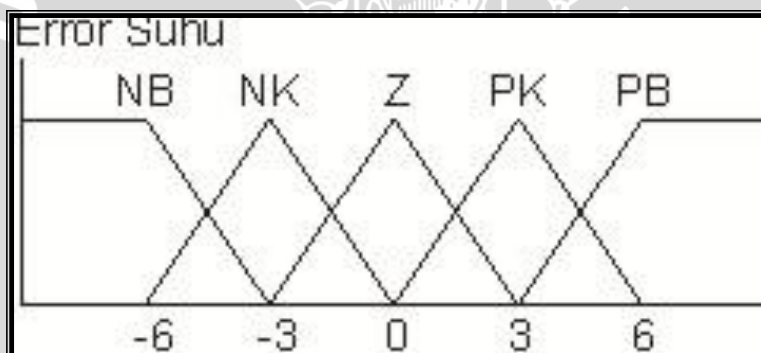
$Error(t-1)$ = *Error* pada waktu (t-1)

4.4.2. Fungsi Keanggotaan Masukan

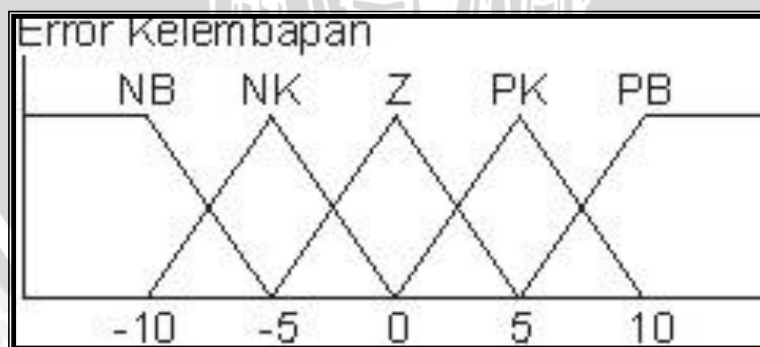
Fungsi keanggotaan dari *Error* dan *delta Error* terdiri dari lima label, yaitu *Negative Big* (NB), *Negative Small* (NS), *Zero* (Z), *Positive Small* (PS), *Positive Big* (PB). *Error* adalah nilai *set point* dikurangi nilai sebenarnya, sedangkan *deltaError* didapat dari *Error* nilai suhu dan kelembaban sekarang dikurangi dengan *Error* nilai suhu dan kelembaban yang sebelumnya. Data masukan dari fungsi keanggotaan didapat dari nilai suhu dan kelembaban yang dikeluarkan oleh sensor *SHT11*.

Error dan delta *Error* suhu memiliki rentang antara -6 hingga +6, karena kondisi *Error* lebih dari nominal yang ditentukan jarang ditemui. Kisaran temperature ini berdasarkan asumsi suhu dimalang jarang melebihi 30°C. *Error* dan delta *Error* kelembaban memiliki rentang antara -10 hingga +10, karena rentang maksimal dari kelembaban adalah 100%RH. Untuk menurunkan suhu membutuhkan waktu yang cukup lama, sehingga diberikan rentang yang kecil.

Label masing-masing fungsi keanggotaan masukan berjumlah 5 dengan tujuan agar rule yang dihasilkan semakin banyak. Semakin banyak rule maka semakin banyak juga kemungkinan kondisi yang tercipta untuk mencapai keluaran yang diinginkan. Disini saya mengambil 5 fungsi keanggotaan karena sudah dirasa optimal, fleksibel, dan tidak terlalu lambat untuk kasus rule *Error* dan delta *Error*. *Membership function* yang digunakan dapat dilihat di Gambar 4.10 dan Gambar 4.11.



Gambar 4.10 Fungsi Keanggotaan Masukan *Error* dan *Delta Error* Suhu



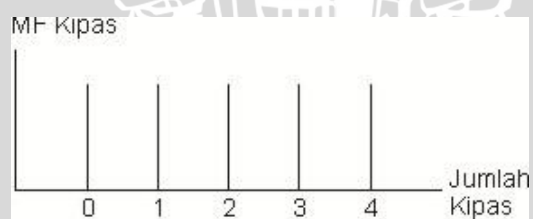
Gambar 4.11 Fungsi Keanggotaan Masukan *Error* dan *Delta Error* Kelembaban

4.4.3. Fungsi Keanggotaan Keluaran

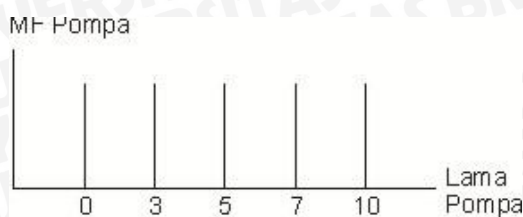
Keluaran dari alat ini menggunakan kipas dan spray yang mana mempunyai aturan yang berbeda. Nilai crisp dari *Error* suhu dan kelembaban diubah untuk digunakan sebagai masukan dari *fuzzy*. Batasan fungsi keanggotaan yang digunakan untuk mengendalikan masukan suhu dan masukan kelembaban adalah besarnya *Error* yang terjadi, untuk *Error* suhu besarnya -6 hingga +6 dengan nilai set point 22°C , sedangkan untuk *Error* kelembaban besarnya -10 hingga +10 dengan nilai set point 90%RH.

Semua batasan ini dimasukkan kedalam suatu fungsi keanggotaan atau *membership function* (MF), baik itu MF *Error* suhu maupun MF *Error* kelembaban. Setelah itu, nilai MF masukan dari *Error* suhu dan *Error* kelembaban dimasukkan kedalam aturan *fuzzy*, yang akan memberikan masukan kepada aktuator, yaitu aturan untuk mengatur suhu dengan kipas, dan aturan untuk mengatur kelembaban dengan pompa air.

Bentuk dari fungsi yang mengatur aktuator adalah singular, yaitu berbentuk diskrit, untuk memudahkan penyusunan algoritma *fuzzy* dan mempercepat proses defuzzifikasi, dan komponen yang dikontrol disini adalah suhu dan kelembaban dimana kepresisiannya tidak dibutuhkan terlalu tinggi. Jumlah dari masing-masing MF ini adalah 5 label dan beserta kondisi yang mewakili, dengan kondisi ini dapat menangani reaksi plant yang lambat dan plant tidak terlalu pengaruh dengan perubahan yang sangat kecil. Lima label dapat dilihat di Gambar 4.12 dan Gambar 4.13.



Gambar 4.12 Fungsi Keanggotaan Keluaran *Error* dan Delta *Error* Suhu



Gambar 4.13 Fungsi Keanggotaan Keluaran *Error* dan *Delta Error* Kelembaban

4.4.4. Perancangan Aturan Fuzzy

Rule (aturan) *fuzzy* merupakan suatu bentuk aturan relasi/implikasi *IF-THEN*, digunakan sebagai penentu keluaran dari inferensi yang akan diolah dalam proses defuzzifikasi, dengan jumlah 2 rule, masing - masing 25 macam. *Rule* tersebut didapat dari 5 *membership function Error* dan 5 *membership function delta Error*.

Rule untuk suhu akan menentukan jumlah kipas yang menyala, sedangkan rule untuk kelembaban akan menentukan lama *sprayer* menyala. Semakin besar *Error* suhu, maka jumlah kipas yang menyala adalah maksimal, yaitu 4 buah, agar *Error* cepat berkurang. Semakin besar *Error* kelembaban, maka *sprayer* akan menyala sangat lama, yaitu 10 detik, agar *Error* cepat berkurang. *Rule fuzzy* yang dipakai dapat dilihat di Tabel 4.2.

Tabel 4.2 Aturan Fuzzy (pengaturan kipas dan pompa)
Sumber: Prasetya, 2015

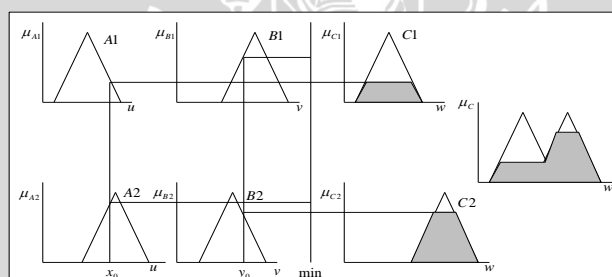
de \ e		kipas				
		NB	NK	Z	PK	PB
NB	3	2	1	0	0	
NK	4	3	2	1	0	
Z	4	4	3	2	1	
PK	4	4	4	2	1	
PB	4	4	4	3	1	

pompa

e \ de	NB	NK	Z	PK	PB
NB	0	0	3	5	5
NK	0	3	5	7	7
Z	0	5	7	7	10
PK	0	5	7	10	10
PB	3	5	10	10	10

4.4.5. Inferensi Min-Max

Inferensi adalah proses evaluasi *rule fuzzy* untuk menghasilkan output dari tiap *rule*. Lebih jelas metode min-max dapat dideskripsikan dalam gambar 4.14 sebagai berikut :



Gambar 4.14 Inferensi Fuzzy dengan Metode MIN-MAX

Untuk mengetahui metode ini pada sistem, perlu melakukan beberapa perhitungan secara manual. Perhitungan hanya dilakukan pada beberapa hasil pengujian keseluruhan sistem. Berikut hasil perhitungan dan Tabel 4.3 yang menampilkan set point dan kondisi suhu dan kelembaban saat pengujian keseluruhan.

Tabel 4.3 set point dan kondisi suhu kelembaban saat pengujian keseluruhan

No	Set Point Suhu (°C)	Set Point Kelembaban (%RH)	Suhu Ruangan (°C)	Kelembaban Ruangan (%RH)
1	22	90	27.89	59.04
2	22	90	22.4	90.01
3	22	90	21.9	87.45

1. Inferensi Min-Max data ke-1

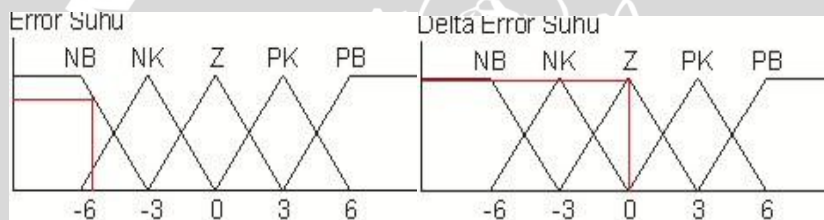
• Suhu

$$\begin{aligned} Error(t) &= SP - PV(t) \\ &= 22 - 27.89 \\ &= -5.89 \end{aligned}$$

$$\begin{aligned} Error(t+1) &= SP - PV(t+1) \\ &= 22 - 27.9 \\ &= -5.9 \end{aligned}$$

$$\begin{aligned} \Delta Error &= Error(t+1) - Error(t) \\ &= -5.9 - (-5.89) \\ &= -0.01 \end{aligned}$$

Nilai derajat keanggotaan suhu dari data ke-1 dapat dilihat di Gambar 4.15.



Gambar 4.15 Inferensi Min-Max data suhu ke-1

Nilai derajat keanggotaan *Error*:

$$Error1 = 0.9 \text{ [NB]}$$

$$Error2 = 0.1 \text{ [NK]}$$

Nilai derajat keanggotaan *DelError*:

$$\Delta Error = 1 \text{ [Z]}$$

• Kelembaban

$$\begin{aligned} Error(t) &= SP - PV(t) \\ &= 90 - 59.04 \\ &= 30.96 \end{aligned}$$

$$\begin{aligned} Error(t+1) &= SP - PV(t+1) \\ &= 90 - 59.01 \\ &= 30.99 \end{aligned}$$

$$\begin{aligned} \Delta Error &= Error(t+1) - Error(t) \\ &= 30.99 - 30.96 \\ &= 0.03 \end{aligned}$$

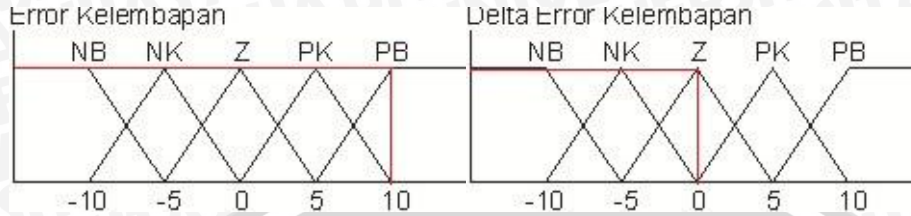
Hasil:

- Jika *Error* NB & *Delta Error* Z, maka \sum kipas = 4 [0.9]

- Jika *Error* NK & *Delta Error* Z, Maka \sum kipas = 4 [0.1]



Nilai derajat keanggotaan kelembaban dari data ke-1 dapat dilihat di Gambar 4.16.



Gambar 4.16 Inferensi Min-Max data kelembaban ke-1

Nilai derajat keanggotaan *Error*:

$$Error = 1 [PB]$$

Nilai derajat keanggotaan *DelError*:

$$Delta Error = 1 [Z]$$

Hasil:

Jika *Error* PB & *Delta Error* Z,
maka lama pompa = 10 [1]

2. Inferensi Min-Max data ke-2

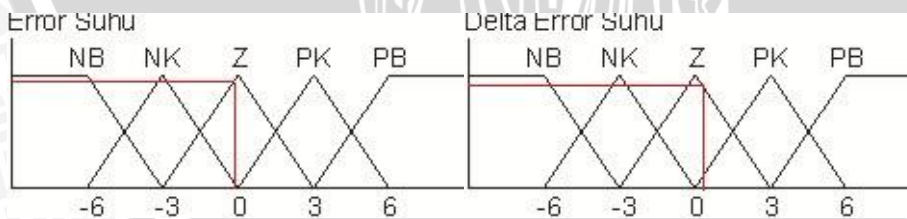
- Suhu

$$\begin{aligned} Error(t) &= SP - PV(t) \\ &= 22 - 22.4 \\ &= -0.4 \end{aligned}$$

$$\begin{aligned} Error(t+1) &= SP - PV(t+1) \\ &= 22 - 22.3 \\ &= -0.1 \end{aligned}$$

$$\begin{aligned} Delta Error &= Error(t+1) - Error(t) \\ &= -0.1 - (-0.4) \\ &= 0.3 \end{aligned}$$

Nilai derajat keanggotaan suhu dari data ke-2 dapat dilihat di Gambar 4.17.



Gambar 4.17 Inferensi Min-Max data suhu ke-2

Nilai derajat keanggotaan *Error*:

$$Error1 = 0.9 [Z]$$

$$Error2 = 0.1 [NK]$$

Nilai derajat keanggotaan *DelError*:

$$Delta Error1 = 0.85 [Z]$$

$$Delta Error2 = 0.15 [PK]$$

Hasil:

-Jika *Error Z & Delta Error*

$$Z, \text{ maka } \sum \text{kipas} = 0.85[3]$$

-Jika *Error Z & Delta Error*

$$PK, \text{ maka } \sum \text{kipas} = 0.15[4]$$

-Jika *Error NK & Delta*

$$Error Z, \text{ maka } \sum \text{kipas} = 0.1[4]$$

-Jika *Error NK & Delta*

$$Error PK, \text{ maka } \sum \text{kipas} = 0.1[4]$$

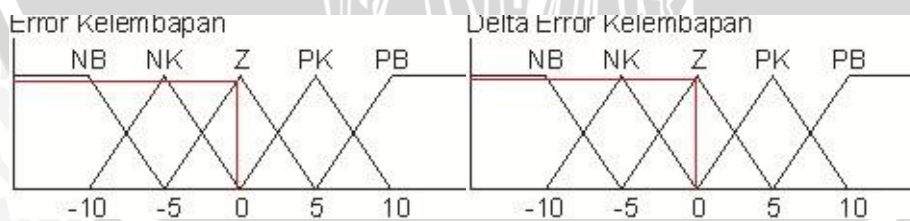
- Kelembaban

$$\begin{aligned} Error(t) &= SP - PV(t) \\ &= 90 - 90.01 \\ &= 0 \end{aligned}$$

$$\begin{aligned} Error(t+1) &= SP - PV(t+1) \\ &= 90 - 90.09 \\ &= -0.09 \end{aligned}$$

$$\begin{aligned} Delta Error &= Error(t+1) - Error(t) \\ &= -0.09 - 0 \\ &= -0.09 \end{aligned}$$

Nilai derajat keanggotaan kelembaban dari data ke-2 dapat dilihat di Gambar 4.18.



Gambar 4.18 Inferensi Min-Max data kelembaban ke-2



Nilai derajat keanggotaan *Error*:

$$Error1 = 0.95 [Z]$$

$$Error2 = 0.05 [NK]$$

Nilai derajat keanggotaan *DelError*:

$$Delta Error1 = 0.95 [Z]$$

$$Delta Error2 = 0.05 [NK]$$

Hasil:

-Jika *Error Z & Delta Error Z*,
maka lama pompa = 0.95[7]

-Jika *Error Z & Delta Error NK*,
maka lama pompa = 0.05[5]

-Jika *Error NK & Delta Error Z*,
maka lama pompa = 0.05[5]

-Jika *Error NK & Delta Error NK*,
maka lama pompa = 0.05[3]

3. Inferensi Min-Max data ke-3

- Suhu

$$Error(t) = SP - PV(t)$$

$$= 22 - 21.9$$

$$= 0.1$$

$$Error(t+1) = SP - PV(t+1)$$

$$= 22 - 21.8$$

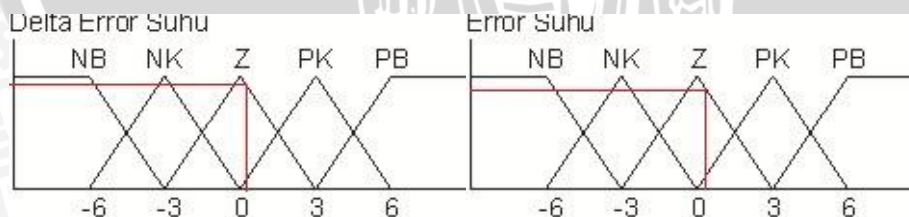
$$= 0.2$$

$$Delta Error = Error(t+1) - Error(t)$$

$$= 0.2 - 0.1$$

$$= 0.1$$

Nilai derajat keanggotaan suhu dari data ke-3 dapat dilihat di Gambar 4.19.



Gambar 4.19 Inferensi Min-Max data suhu ke-3

Nilai derajat keanggotaan *Error*:

$$Error1 = 0.9 [Z]$$

$$Error2 = 0.1 [PK]$$

Nilai derajat keanggotaan *DelError*:

$$Delta Error1 = 0.95 [Z]$$

$$Delta Error2 = 0.05 [PK]$$

Hasil:

-Jika *Error Z* & *Delta Error Z*,
maka \sum kipas = 0.9[3]

-Jika *Error Z* & *Delta Error PK*,
maka \sum kipas = 0.05[4]

-Jika *Error PK* & *Delta Error Z*,
maka \sum kipas = 0.1[2]

-Jika *Error PK* & *Delta Error PK*,
maka \sum kipas = 0.05[2]

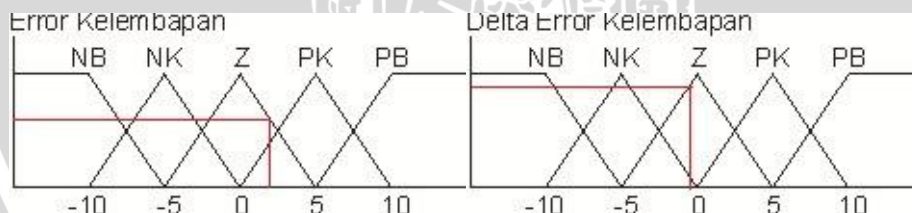
- Kelembaban

$$\begin{aligned} Error(t) &= SP - PV(t) \\ &= 90 - 87.45 \\ &= 2.55 \end{aligned}$$

$$\begin{aligned} Error(t+1) &= SP - PV(t+1) \\ &= 90 - 87.62 \\ &= 2.38 \end{aligned}$$

$$\begin{aligned} Delta Error &= Error(t+1) - Error(t) \\ &= 2.38 - 2.55 \\ &= -0.17 \end{aligned}$$

Nilai derajat keanggotaan kelembaban dari data ke-3 dapat dilihat di Gambar 4.20.



Gambar 4.20 Inferensi Min-Max data kelembaban ke-3

Nilai derajat keanggotaan *Error*:

$$Error1 = 0.55 [Z]$$

$$Error2 = 0.45 [PK]$$

Nilai derajat keanggotaan *DelError*:

$$Delta Error1 = 0.9 [Z]$$

$$Delta Error2 = 0.1 [NK]$$

Hasil:

-Jika *Error Z & Delta Error Z*,
maka lama pompa = 0.55[7]

-Jika *Error Z & Delta Error NK*,
maka lama pompa = 0.1[5]

-Jika *Error PK & Delta Error Z*,
maka lama pompa = 0.45[7]

-Jika *Error PK & Delta Error NK*,
maka lama pompa = 0.1[7]

Data Tabel 4.3 diambil dari data pengujian alat secara keseluruhan secara acak. Untuk nilai *Error* (t+1) juga didapatkan dari data pengujian alat secara keseluruhan. Pada data pertama kelembaban, ditemukan *Error* sebesar 30.96%RH, nilai *Error* tersebut sangat jauh diatas rentang fungsi keanggotaan masukan, karena kondisi seperti ini sangat jarang ditemui. Melalui proses inferensi Min-Max didapat nilai derajat keanggotaan sebesar 1[PB], dan hasilnya *sprayer* akan menyala selama 10 detik. Disini kontroler terbukti dapat mengatasi kondisi tersebut, karena di data ketiga dan seterusnya kelembaban terus naik.

4.4.6. Defuzzifikasi

Defuzzifikasi adalah proses untuk mengubah keluaran *fuzzy* menjadi keluaran *crisp*. Hal ini diperlukan karena dalam aplikasi nyata dibutuhkan sebuah nilai tegas (*crisp*). Hasil defuzzifikasi inilah yang digunakan untuk mengatur keluaran kipas dan spray. Metode defuzzifikasi yang digunakan adalah *Max-membership principle* untuk suhu dan *Weighted Average* untuk kelembaban. *Max-membership principle* mengambil nilai fungsi keanggotaan terbesar dari keluaran inferensi yang ada untuk dijadikan nilai defuzzifikasi, pemilihan metode ini karena keluaran dari suhu adalah jumlah kipas yang menyala, sehingga dibutuhkan nilai yang bulat. Sedangkan metode *Weighted Average* mengambil

nilai rata-rata dengan menggunakan pembobotan berupa derajat keanggotaan. Sehingga y' didefinisikan sebagai :

$$y' = \sum \frac{\mu(y)y}{\mu(y)} \quad (4.3)$$

Dimana y adalah nilai crisp dan $\mu(y)$ adalah derajat keanggotaan dari nilai *crisp* y . berdasarkan contoh hasil proses inferensi, maka keluaran untuk *fan* dan *sprayer* didapatkan:

1. Defuzzifikasi untuk inferensi Min-Max saat data ke-1

- Suhu

$$0.9[4] \vee 0.1[4] \rightarrow 4$$

- Kelembaban

$$\begin{aligned} y' &= \sum \frac{\mu(y)y}{\mu(y)} \\ &= \frac{1 \times (10)}{1} \\ &= \frac{10}{1} \\ &= 10 \end{aligned}$$

2. Defuzzifikasi untuk inferensi Min-Max data ke-2

- Suhu

$$0.85[3] \vee 0.15[4] \vee 0.1[4] \vee 0.1[4] \rightarrow 3$$

- Kelembaban

$$\begin{aligned} y' &= \sum \frac{\mu(y)y}{\mu(y)} \\ &= \frac{(0.95 \times [7]) + (0.05 \times [5]) + (0.05 \times [5]) + (0.05 \times [3])}{0.95 + 0.05 + 0.05 + 0.05} \\ &= \frac{6.65 + 0.25 + 0.25 + 0.15}{1.2} \\ &= 6.08 \end{aligned}$$

3. Defuzzifikasi untuk inferensi Min-Max data ke-3

- Suhu

$$0.9[3] \vee 0.05[4] \vee 0.1[2] \vee 0.05 [2] \rightarrow 3$$

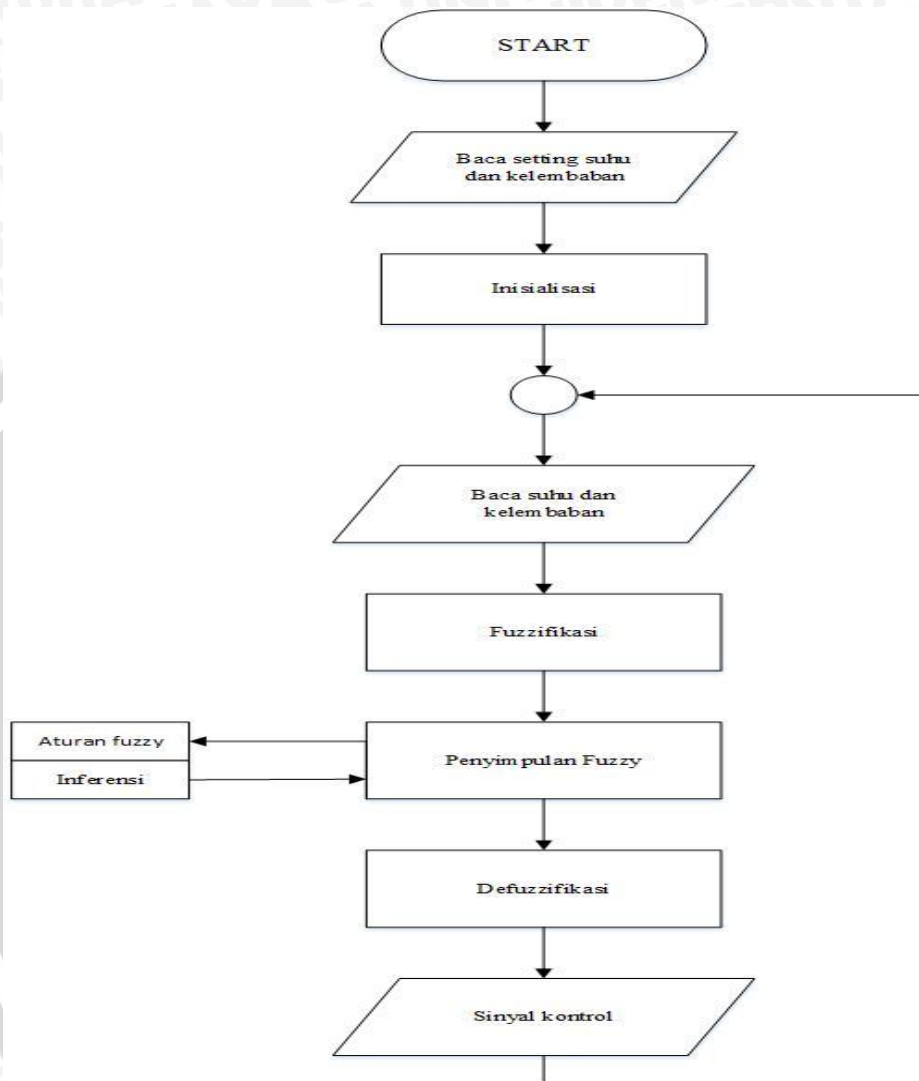
- Kelembaban

$$\begin{aligned} y' &= \sum \frac{\mu(y)y}{\mu(y)} \\ &= \frac{(0.55 \times [7]) + (0.1 \times [5]) + (0.45 \times [7]) + (0.1 \times [7])}{0.55 + 0.1 + 0.45 + 0.1} \\ &= \frac{3.85 + 0.5 + 3.15 + 0.7}{1.2} \\ &= 6.833 \end{aligned}$$

Dari hasil perhitungan defuzzifikasi diatas, dapat disimpulkan bahwa sistem sudah mampu menjalankan logika *fuzzy*. Hasil defuzzifikasi bukan bilangan bulat karena perhitungan masih dilakukan manual. Saat didalam program, keluaran yang diperintahkan kepada aktuator sudah bilangan bulat karena didalam pemrograman hasil perhitungan defuzzifikasi sudah dibulatkan oleh integer.

4.5. Perancangan Perangkat Lunak

Flowchart perancangan perangkat lunak ditunjukkan dalam Gambar 4.21 berikut



Gambar 4.21 Flowchart Program

BAB V

PENGUJIAN & 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.

5.1. Pengujian Rangkain Sensor Suhu dan Kelembaban SHT11

Seperti yang diketahui bahwa sensor *SHT11* memiliki protokol komunikasi I²C, sehingga untuk melakukan pengambilan datanya adalah dengan pengaturan pin SDA dan SCL untuk meminta sinyal *acknowledge* dari *SHT11*, data. Sebagai perbandingan dari hasil kelembaban, akan dibandingkan dengan alat pengukur suhu dan kelembaban yang sudah jadi.

5.1.1. Prosedur Pengujian

1. Menghubungkan *board* sensor suhu dan kelembaban SHT11 pada PORTB mikrokontroler
2. Menyalakan *power supply*
3. Melihat dan bandingkan data pada LCD dengan alat pengukur suhu yang sudah jadi
4. Mencatat data dan mengulangi sebanyak 10 kali dengan *interval* pengambilan data random.

Prosedur pengujian akurasi sensor SHT11 dapat dilihat pada Gambar 5.1.



Gambar 5.1 Pengujian Akurasi Sensor SHT11

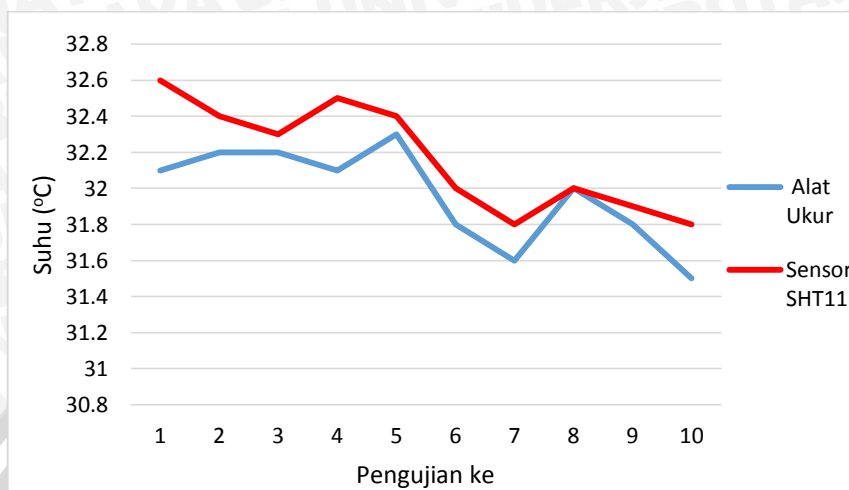
5.1.2. Hasil Data Pengujian

Data hasil pengujian sensor SHT11 dapat dilihat pada Tabel 5.1.

Tabel 5.1 Data Hasil Pengujian Akurasi Sensor Suhu dan Kelembaban SHT11

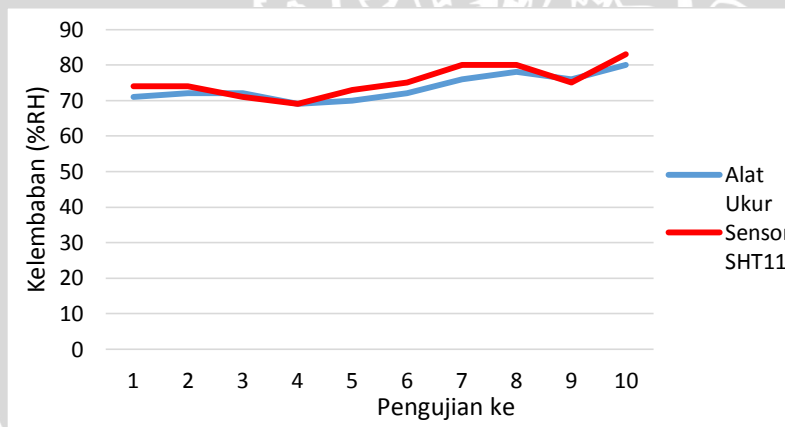
No	AlatUkur	Sensor		AlatUkur	Sensor	
	(°C)	(°C)	%Err	(%RH)	(%RH)	%Err
1	32.10	32.6	1.56	71	74	4.23
2	32.20	32.4	0.62	72	74	2.78
3	32.20	32.3	0.31	72	71	1.39
4	32.10	32.5	1.25	69	69	0.00
5	32.30	32.4	0.31	70	73	4.29
6	31.8	32	0.63	72	75	4.17
7	31.6	31.8	0.63	76	80	5.26
8	32.0	32	0.00	78	80	2.56
9	31.8	31.9	0.31	76	75	1.31
10	31.5	31.8	0.95	80	83	3.75
Rata – Rata		32.17	0.66		75.40	2.97

Grafik pengujian sensor suhu *SHT11* dapat dilihat pada Gambar 5.2



Gambar 5.2 Grafik hasil pengujian sensor suhu *SHT11*

Gambar 5.3 adalah grafik pengujian kelembaban pada sensor *SHT11*



Gambar 5.3 Grafik hasil pengujian sensor kelembaban *SHT11*

5.1.3. Analisa dan Kesimpulan

Dari pengujian yang telah dilakukan, didapatkan data seperti pada tabel 5.1. Pada sensor SHT11 ini mempunyai selisih *error* yang kecil dan mendekati sama pada tiap percobaan.

Dari data yang telah didapatkan dalam pengujian ini, seperti yang terlihat pada tabel 5.1 nilai rata-rata *error* dari sensor untuk suhu adalah 0.66% dengan *error* terbesar

1.56%. Untuk *error* kelembaban rata-rata dari sensor adalah 2.97%, dengan nilai kesalahan terbesar 5.26%. *Error* yang terjadi antara sensor dengan acuan dapat terjadi bisa dikarenakan perbedaan posisi sensor, serta pergerakan udara yang diterima oleh sensor, serta bisa disebabkan oleh kualitas dari sensor dan alat ukur yang berbeda.

Dari hasil pengujian akurasi sensor ini dapat disimpulkan bahwa nilai pembacaan suhu dan kelembaban udara antara sensor SHT11 dengan acuan mempunyai perbedaan yang kecil, untuk suhu nilai rata-rata kesalahan yaitu 0.66% sedangkan untuk rata-rata kesalahan kelembaban yaitu sebesar 2.97%.

5.2. Pengujian Respon alat

5.2.1. Pengujian Respon Kontrol Suhu

Pengujian ini dilakukan untuk mengetahui respon sistem kontrol suhu pada alat yang telah dibuat. Membandingkan suhu didalam dan diluar kumbung jamur.

a. Prosedur Pengujian

- 1.Menghubungkan board sensor pada PORTB mikrokontroler.
- 2.Mematikan kontrol kelembaban.
- 3.Menyalakan power supply.
- 4.Mencatat data suhu luar dan dalam ruang.

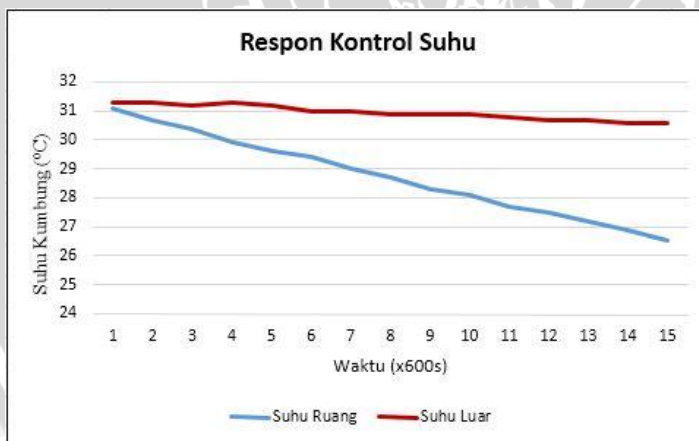
b. Hasil Data Pengujian

Data pengujian respon kontrol suhu dapat dilihat pada Tabel 5.2.

Tabel 5.2 Data Pengujian Respon Kontrol Suhu

No	Jam	Suhu Ruang (°C)	Suhu Luar (°C)
1	14:14	31.1	31.3
2	14:20	30.7	31.3
3	14:30	30.4	31.2
4	14:40	29.9	31.3
5	14:50	29.6	31.2
6	15:00	29.4	31.0
7	15:10	29	31.0
8	15:20	28.7	30.9
9	15:30	28.3	30.9
10	15:40	28.1	30.9
11	15:50	27.7	30.8
12	15:00	27.5	30.7
13	15:10	27.2	30.7
14	15:20	26.9	30.6
15	15:30	26.5	30.6

Grafik hasil pengujian respon kontrol suhu ditunjukkan pada Gambar 5.4.



Gambar 5.4 Grafik hasil pengujian respon kontrol suhu

c. Analisa dan Kesimpulan

Pengujian ini dimulai pada pukul 14.00, dengan keadaan diluar ruang cerah dan berawan. Suhu yang tercatat sebelum pengukuran yaitu 31.3°C. Pengujian dilakukan

sebanyak 15 kali dengan interval pengambilan data tiap 10 menit sekali. *Set point* yang diberikan 27°C , hanya saat pengujian respon suhu saja.

Dari data tabel 5.2 menunjukkan penurunan suhu dalam kumbung dengan awal 31.1°C dan suhu akhir pengukuran 26.5°C . perbedaan suhu luar dan dalam ruang pada awal pengukuran sebesar 0.2°C dan pada akhir pengukuran sebesar 4.1°C .

Penurunan suhu ini sudah sesuai dengan harapan. Meskipun saat pengujian cuaca sangat panas dan waktu pengujian yang sebentar. Tetapi dari grafik 5.4 dapat dilihat perbedaan suhu luar dan dalam ruang meningkat, bila pengujian diteruskan dapat dipastikan suhu didalam kumbung jamur akan jauh berbeda dengan suhu diluar kumbung jamur.

5.2.2. Pengujian Respon Kontrol Kelembaban

Pengujian ini dilakukan untuk mengetahui respon sistem kontrol kelembaban pada alat yang telah dibuat. Membandingkan kelembaban diluar dan didalam kumbung jamur.

a. Prosedur Pengujian

1. Menghubungkan board sensor pada PORTB mikrokontroler.
2. Mematikan kontrol suhu.
3. Menyalakan power supply.
4. Mencatat data kelembaban luar dan dalam ruang.

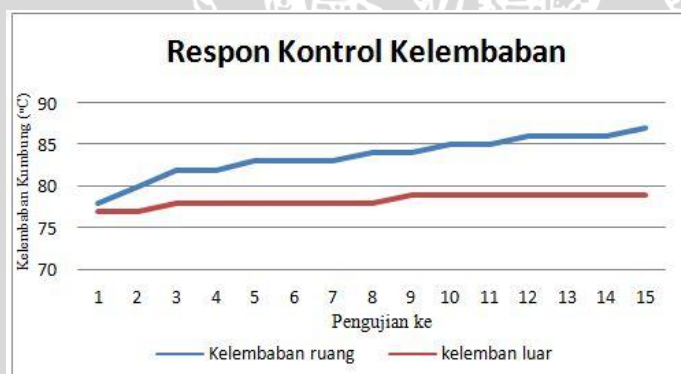
b. Hasil Data Pengujian

Data pengujian respon kontrol kelembaban dapat dilihat pada Tabel 5.3.

Tabel 5.3 Data Pengujian Respon Kontrol Kelembaban

No	Jam	Kelembaban Ruang (%RH)	Kelembaban Luar (%RH)
1	15:40	78	77
2	15:50	80	77
3	15:00	82	78
4	16:10	82	78
5	16:20	83	78
6	16:30	83	78
7	16:40	83	78
8	16:50	84	78
9	17:00	84	79
10	17:10	85	79
11	17:20	85	79
12	17:30	86	79
13	17:40	86	79
14	17:50	86	79
15	18:00	87	79

Grafik hasil pengujian respon kelembaban ditunjukkan pada gambar 5.5.



Gambar 5.5 Grafik hasil pengujian respon kontrol kelembaban

c. Analisa dan Kesimpulan

Pengujian ini dilakukan setelah pengujian respon suhu, dengan keadaan luar lebih mendung dari saat pengujian respon suhu sebelumnya. Kelembaban yang tercatat sebelum pengujian dilakukan yaitu sebesar 78%. *Set point* yang diberikan sebesar 80%, hanya saat pengujian respon kelembaban saja.

Pada kumbung jamur ini terdapat media tanah sebagai pelembab. Pada saat pengujian kondisi tanah sedikit basah, namun terdapat tanah yang kering di beberapa bagian kumbung. Pada tabel 5.3, pada pengukuran pertama tercatat nilai kelembaban 78%, kontrol merespon dengan melakukan penyiraman di dalam kumbung. Pada pengukuran kedua dikarenakan kondisi kumbung sudah mencapai nilai *set point* yaitu 80%, keluaran dari fuzzy menjadi 0, karena selama pengukuran kelembaban tidak turun. Dikarenakan kondisi tanah yang basah, kelembaban terus meningkat hingga 87%RH.

Dari penjabaran di atas dapat disimpulkan bahwa respon kontrol kelembaban sudah sesuai dengan target yang diharapkan, karena dapat mencapai *set point*. Kontrol dapat segera merespon ketika ada perubahan yang terjadi ketika kelembaban berada dibawah nilai *set point*.

5.3. Pengujian Alat Keseluruhan

Pengujian ini dilakukan untuk mengetahui respon alat secara keseluruhan dengan melakukan pengambilan data pada ruang budidaya.

5.3.1. Prosedur Percobaan

1. Menghubungkan *board* sensor suhu dan kelembaban SHT11 pada PORTB mikrokontroler
2. Menyalakan power *supply*
3. waktu *sampling* sebesar 20 detik
4. Mencatat data dan ulangi sebanyak 15 kali

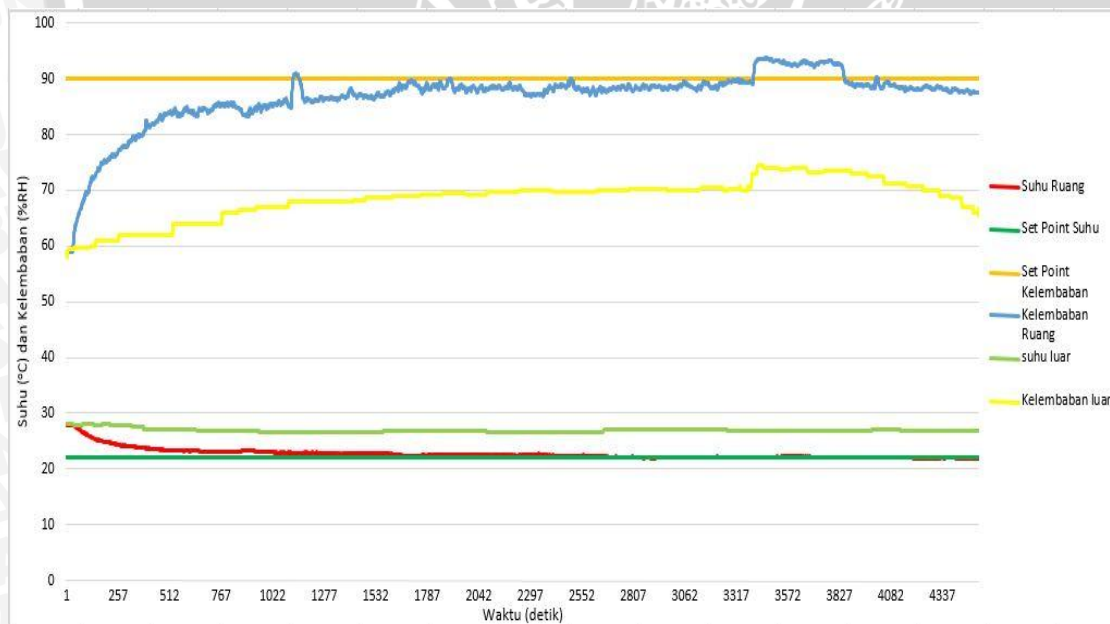
5.3.2. Hasil Percobaan

Data pengujian keseluruhan dapat dilihat pada Tabel 5.4.

Tabel 5.4 Data Pengujian Keseluruhan

No	Suhu		Kelembaban	
	Suhu Ruang (°C)	Fan yang menyala (0-4)	RH Ruang (%RH)	T _{on} (detik)
1	27.89	4	59.04	10
2	24.12	4	77.71	10
3	23.24	3	84.26	7
4	23.33	3	83.38	7
5	22.86	3	86.29	7
6	22.79	3	86.80	7
7	22.46	3	88.30	7
8	22.46	3	88.74	7
9	22.37	3	88.17	7
10	22.11	2	88.46	7
11	22.13	2	88.55	7
12	21.98	2	89.55	5
13	22.26	3	92.58	3
14	22.03	2	88.94	5
15	21.90	2	88.28	7

Grafik hasil pengujian keseluruhan dapat dilihat pada Gambar 5.6.



Gambar 5.6 Grafik hasil pengujian keseluruhan

5.3.3. Analisa dan Kesimpulan

Dalam pengujian alat secara keseluruhan ini sistem berjalan selama sekitar 2 jam, dengan kondisi cuaca cerah berawan. Pengambilan data yang terdapat pada Tabel 5.4 adalah pengambilan data secara acak sebanyak 15 kali.

Kondisi diawal pengukuran menunjukkan suhu kumbung 27.89°C dengan kelembaban $59.04\% \text{RH}$. Dari data pada Gambar 5.6, terlihat grafik kelembaban mengalami *overshoot* pada detik ke 1100, kemudian mencapai kondisi *steady state* pada detik ke 1600. Nilai dari *overshoot* sebesar 0.8% , jauh di bawah nilai toleransi sebesar 5% . Setelah detik ke 1600, nilai dari kelembaban sudah mulai stabil, dengan *error* maksimal 3.27% . Pada detik ke 3392-3875, nilai grafik kelembaban mengalami kenaikan hingga melebihi *set point*, hal ini dikarenakan kondisi cuaca disaat pengujian tiba-tiba berubah mendung gerimis, kemudian nilai kelembaban kembali ke nilai *set point* dan cenderung turun, tapi tidak sampai melebihi *error* maksimal. Grafik suhu cenderung lebih stabil, mencapai kondisi *steady state* pada detik ke 1100. Nilai suhu terus turun dengan *error* maksimal saat kondisi *steady state* adalah 0.8°C . Saat detik 800-1000, ada sedikit kenaikan pada nilai suhu, hal ini dikarenakan kondisi cuaca yang berubah-ubah.

BAB VI

PENUTUP

6.1. Kesimpulan

Pada penelitian ini telah dirancang sistem pengontrolan suhu dan kelembaban pada miniatur kumbung jamur tiram menggunakan kontrol logika *fuzzy* dengan *set point* suhu sebesar 22°C dan *set point* kelembaban sebesar 90%RH. Hasil pengujian seluruh sistem menggunakan kontrol logika *fuzzy* dengan metode inferensi adalah metode min-max sedangkan pada defuzzifikasi menggunakan metode *max-membership principle* dan *weighted average*, menggunakan 5 *membership function* pada *error* dan *delta error*. *Input fuzzy* berupa nilai suhu dan kelembaban sebagai *error* dan selisih dari nilai *error* suhu sekarang dan sebelumnya sebagai *delta error*. Jumlah *rule* yang dipergunakan berjumlah 25.

Secara umum suhu dan kelembaban yang didapatkan setelah proses kontrol memenuhi kebutuhan jamur tiram untuk tumbuh yaitu berkisar antara $22\text{-}27^{\circ}\text{C}$ dan 80-90%RH. *Error* suhu secara keseluruhan sistem adalah 3.87% dan untuk kelembaban adalah 3.27%. Sistem dapat mencapai target dari *set point* yang telah ditentukan. Hasil ini menunjukkan bahwa dengan kontrol logika *fuzzy* dapat mengontrol suhu dan kelembaban di dalam miniatur kumbung jamur. Hasil produk yang dihasilkan miniatur kumbung jamur memiliki kualitas yang hampir sama dengan yang dihasilkan petani.

6.2. Saran Pengembangan

Dalam perancangan dan pembuatan alat ini masih terdapat kelemahan, untuk pengembangan dan penelitian lebih lanjut disarankan:

1. Membuat kumbung jamur yang adiabatik, sehingga suhu dan kelembaban diluar ruangan tidak mempengaruhi kondisi didalam kumbung jamur.
2. Menggunakan mini ac *portable* sebagai aktuator suhu untuk pendinginan yang lebih baik.

DAFTAR PUSTAKA

- Al Hamam , Baihaqqi. 2013. Rancang Bangun Sistem Kontrol Dan Monitoring Ruang Budidaya Jamur Tiram Berbasis Multidrop Point Rs485, proyek akhir, PENS, Surabaya, Indonesia
- J. Yan., M. Ryan & J. Power. 1994. Using Fuzzy Logic Towards Intelligent Systems. Prentice Hall – Englewood Cliffs
- Killian, Christopher T. 1996. Modern Control Technology Components and Systems. 1st edition.
- Kuswandi, Son. 2000. Kendali Cerdas (Intelligent Control): EEPIS Press
- Maulana Sy., Erie. 2012. Panen Jamur Tiap Musim.Yogyakarta: Lily Publisher
- Seto Irawan, Dimas. 2011. Aplikasi Fuzzy Pada Pengaturan Rumah Kaca Tanaman Dataran Tinggi, proyek akhir, PENS, Surabaya, Indonesia
- Sudrajat, 2008, Dasar-Dasar Fuzzy Logic, modul kuliah jurusan matematika, UNPAD, Bandung, Indonesia
- Teguh Dwi Prasetya, Patriot. 2015. Implementasi Metode Logika Fuzzy Untuk Pengaturan Suhu Dan Kelembaban Kumbung Budidaya Jamur Tiram, proyek akhir, PENS, Surabaya, Indonesia
- www.arduino.cc Diakses 10 April 2015
- www.adafruit.com Diakses 10 April 2015

LAMPIRAN I

Foto Alat





Foto Mekanik Kumbung Jamur

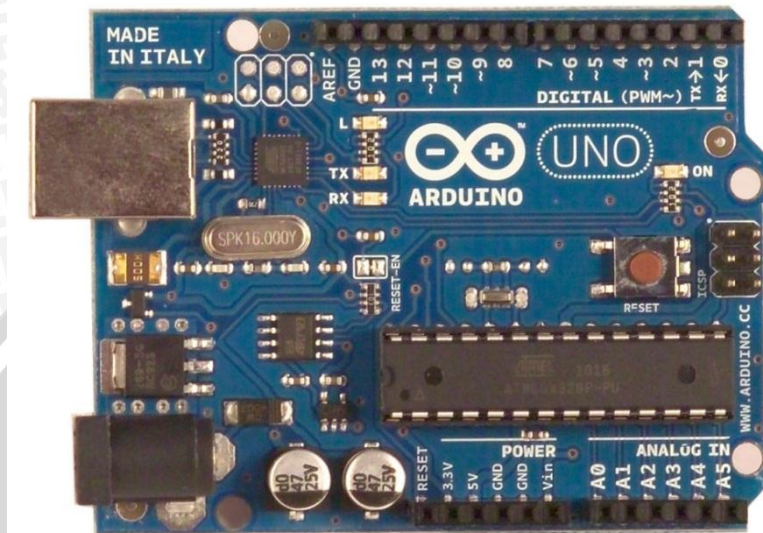
UNIVERSITAS BRAWIJAYA

LAMPIRAN II

Data Sheet



Arduino UNO



Product Overview

The Arduino Uno is a microcontroller board based on the ATmega328 ([datasheet](#)). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, 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 Uno 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.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version

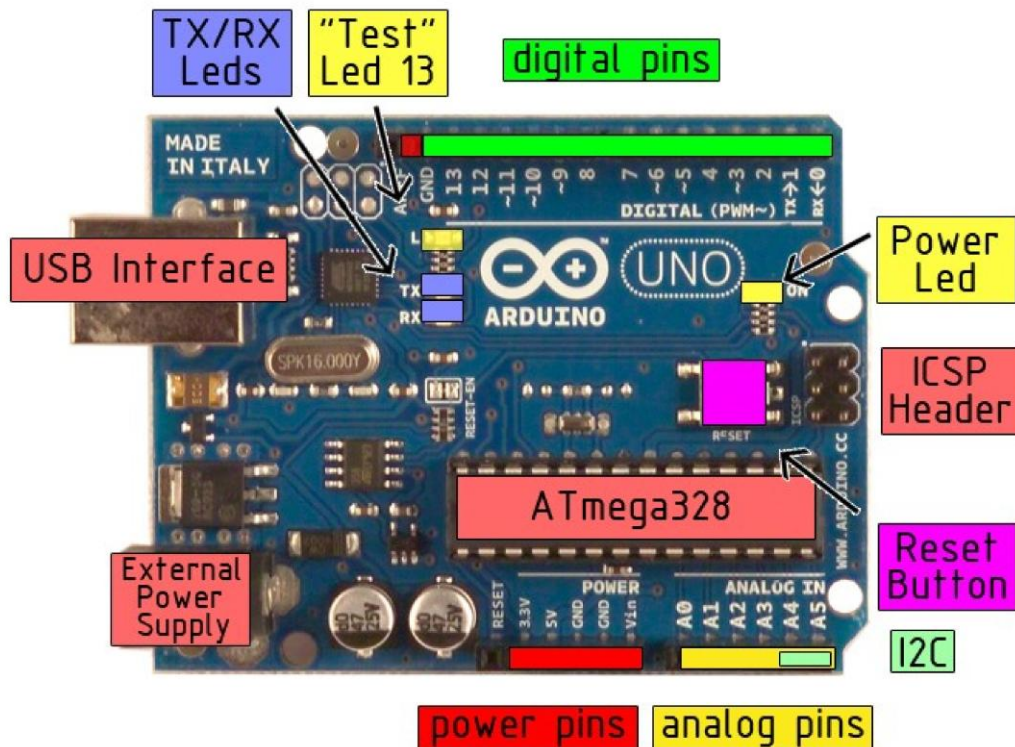
1.1 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB

Arduinoboards, and thereferencemodelfortheArduinoplatform; foracomparisonwithpreviousversions, see the [index of Arduinoboards](#).

Summary

Microcontroller	ATmega328	
Operating Voltage	5V Input Voltage (recommended) 7-12V Input Voltage (limits) 6-20V	
Digital I/O Pins	14 (of which 6 provide PWM output)	
Analog Input Pins	6	
DC Current per I/O Pin	40 mA DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB of which 0.5 KB used by bootloader	
SRAM	2KB	
EEPROM	1KB	
Clock Speed	16MHz	

the board



radiospares

RADIONICS



Power

The Arduino Uno 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 20volts. 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 12volts.

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 powerjack, 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 50mA.
- **GND.** Ground pins.

Memory

The Atmega328 has 32 KB of flash memory for storing code (of which 0,5 KB is used for the bootloader); It has also 2KB of SRAM and 1KB of EEPROM (which can be read and written with the [EEPROM library](#)).

Input and Output

Each of the 14 digital pins on the Uno 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).** Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins in the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts: 2 and 3.** 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: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.
- **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.

The Uno has 6 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 the [analogReference\(\)](#) function. Additionally, some pins have specialized functionality:

- **I²C: 4 (SDA) and 5 (SCL).** Support I²C (TWI) communication using the [Wire library](#).

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 Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega8U2 on the board channels this serial communication over USB and appears as a virtual COM port to software on the computer. The '8U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, an *.inf file is required..

The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

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

The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the [documentation](#) for details. To use the SPI communication, please see the ATmega328 datasheet.

Programming

The Arduino Uno can be programmed with the Arduino software ([download](#)). Select "Arduino Uno w/ ATmega328" from the **Tools > Board** menu (according to the microcontroller on your board). For details, see the [reference](#) and [tutorials](#).



The ATmega328 on the Arduino Uno 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 headerfiles](#)).

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

The ATmega8U2 firmware source code is available . The ATmega8U2 is loaded with a DFU bootloader, which can be activated by connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2. You can then use [Atmel's FLIP software](#) (Windows) or the [DFU programmer](#) (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFUbootloader).



radiospares

RADIONICS



Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno 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 ATmega328 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 Uno 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 Uno. 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 Uno 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 Uno 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 500mA is

applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics



radiospares

RADIONICS



The maximum length and width of the Uno PCB are 2.7 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.

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-platform 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 USBcable.

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>Sketchbo
ok>Arduino-
0017>Example
s>Digital>Blin
k**



radiospares

RADIONICS



ALLIED ELECTRONICS
AN ELECTRONIC COMPONENTS COMPANY



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

```

Blink | Arduino 0017
File Edit Sketch Tools Help
Blink$
int ledPin = 13; // LED connected to digital pin 13

// The setup() method runs once, when the sketch starts


void setup() {
  // initialize the digital pin as an output:
  pinMode(ledPin, OUTPUT);
}


// the loop() method runs over and over again,
// as long as the Arduino has power


void loop()
{
  digitalWrite(ledPin, HIGH); // set the LED on
  delay(1000); // wait for a second
  digitalWrite(ledPin, LOW); // set the LED off
  delay(1000); // wait for a second
}
    
```


In **Tools>Board** select **Arduino Uno**. Now you have to go to

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

 Done compiling.

 Upload

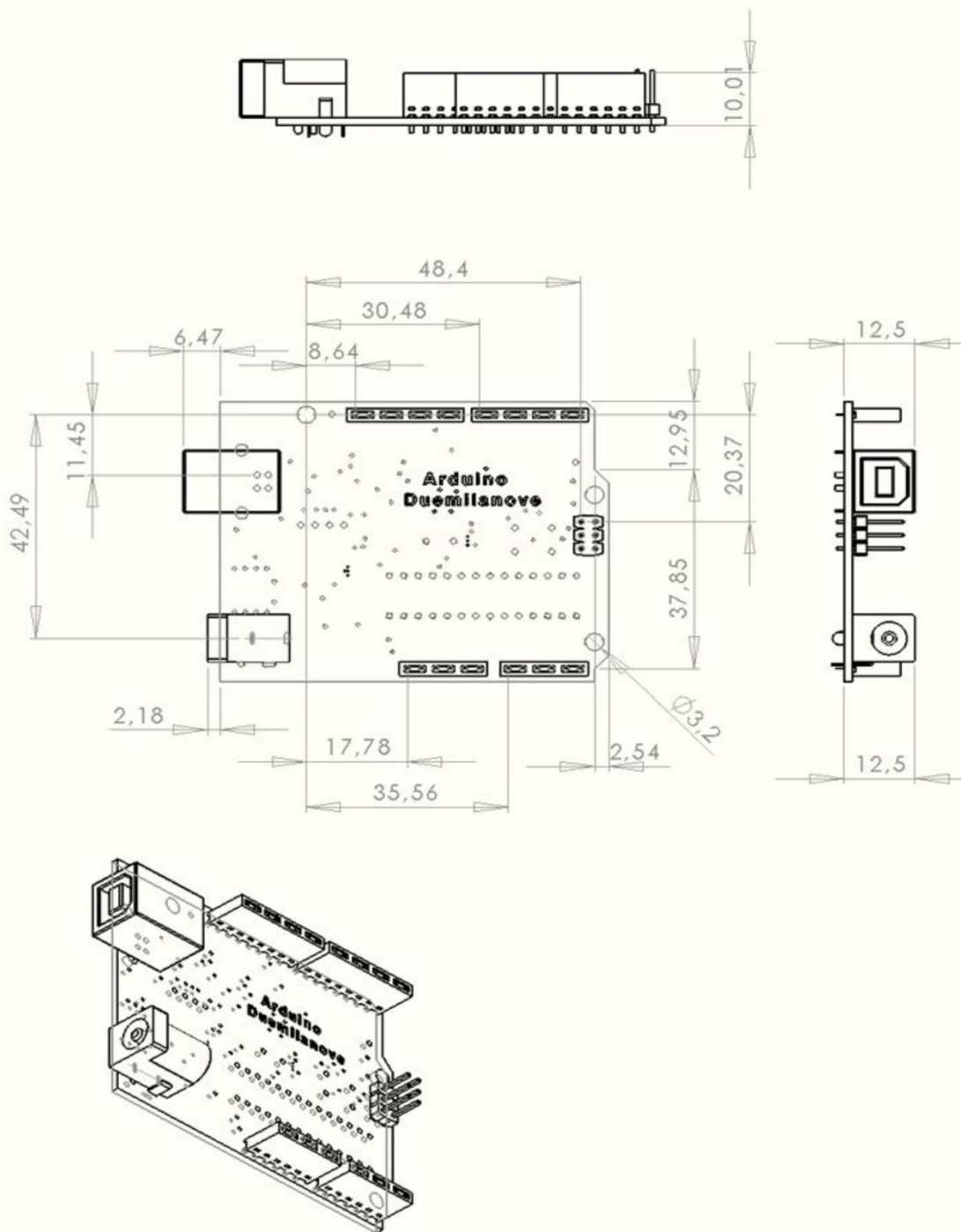
 TX RX Flashing

 Blinking Led!

Press Compile button (to check for errors)



Dimensioned Drawing



radiospares

RADIONICS



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 defect that is the 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 these presentation 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.



Environmental 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.