



**RANCANG BANGUN PENYIRAMAN AIR OTOMATIS DAN  
PROTEKSI HAMA TANAMAN MENGGUNAKAN  
*FUZZY LOGIC CONTROL***

**LAPORAN PROYEK AKHIR**

**Diajukan Kepada Fakultas Teknik Universitas Negeri Yogyakarta  
Untuk Memenuhi Sebagaimana Persyaratan  
Guna Memperoleh Gelar Ahli Madya Teknik**



**Oleh  
SETO AJI PRATAMA  
NIM.15507134010**

**PROGRAM STUDI TEKNIK ELEKTRONIKA  
FAKULTAS TEKNIK  
UNIVERSITAS NEGERI YOGYAKARTA  
2018**

LEMBAR PERSETUJUAN  
PROYEK AKHIR

RANCANG BANGUN PENYIRAMAN AIR OTOMATIS DAN PROTEKSI  
HAMA TANAMAN MENGGUNAKAN FUZZY LOGIC CONTROL

Oleh

SETO AJI PRATAMA

15507134010

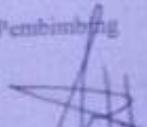
Telah diperiksa dan disetujui Pembimbing

Untuk diujicobakan

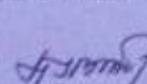
Yogyakarta, 6 Jatis 2018

Menyetujui,

Dosen Pembimbing

  
Dr. Fatchul Anifis, M.T.  
NIP. 19720508 198802 1 002

Mengetahui,  
Kaprodi Teknik Elektronika

  
Dr. Sri Waluyanti, M.Pd.  
NIP. 19581218 198603 2 001

LEMBAR PENGESAHAN

PROYEK AKHIR

RANCANG BANGUN PENYIRAMAN AIR OTOMATIS DAN PROTEKSI  
HAMA TANAMAN MENGGUNAKAN FUZZY LOGIC CONTROL

Dipersiapkan dan Disusun oleh :

**SETO AJI PRATAMA**

15507134010

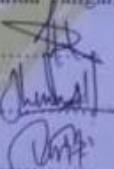
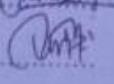
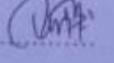
Telah dipertahankan didepan Dewan Penguji Proyek Akhir  
FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA

Pada tanggal 03 Juli 2018

Dan dinyatakan Telah Memenuhi Syarat Guna Memperoleh Gelar

Ahli Madya Teknik

Susunan Dewan Penguji

Jabatan	Nama Lengkap Penguji	Tanda Tangan	Tanggal
Ketua Penguji	Dr. Fatchul Arifin, M.T		18/07/2018
Sekretaris Penguji	Ahmad Awaluddin, M.Pd		18/07/2018
Penguji Utama	Dessy Irmawati, M.T		10/07/2018

Yogyakarta, 03 Juli 2018  
Dekan Fakultas Teknik UNY



Dr. Widarta, M.Pd  
NIP. 19631230 198812 1 001

### LEMBAR PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini :

Nama : Seto Aji Pratama

NIM : 15507134010

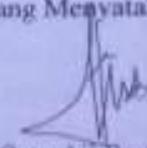
Program Studi : Teknik Elektronika (D3)

Judul Proyek Akhir : Penyiraman Air Otomatis dan Proteksi Hama Tanaman  
menggunakan *Fuzzy Logic Control*

Menyatakan bahwa proyek akhir ini adalah hasil pekerjaan saya sendiri, dan tidak berisi materi yang ditulis oleh orang lain sebagai persyaratan penyelesaian studi di Universitas Negeri Yogyakarta atau perguruan tinggi lainnya, kecuali bagian-bagian tertentu yang saya ambil sebagai acuan dengan mengikuti kaidah penulisan karya ilmiah yang benar. Jika terbukti pernyataan ini tidak benar sepenuhnya menjadi tanggung jawab saya.

Yogyakarta, 3 Juni 2018

Yang Menyatakan,



Seto Aji Pratama

NIM. 15507134010

# **RANCANG BANGUN PENYIRAMAN AIR OTOMATIS DAN PROTEKSI HAMA TANAMAN MENGGUNAKAN FUZZY LOGIC CONTROL**

Oleh : Seto Aji Pratama

NIM : 15507134010

## **ABSTRAK**

Kebutuhan air bagi tanaman merupakan hal yang penting agar tanaman dapat tumbuh dengan subur, tidak hanya itu proteksi tanaman terhadap hama / penyakit tanaman juga penting agar tanaman dapat hidup dan berkembang biak. Namun dalam praktiknya manusia sering lalai dalam melakukan penyiraman air dan proteksi hama tanaman, untuk mengatasi masalah tersebut, Penyiraman Air Otomatis dan Proteksi Hama Tanaman menggunakan *Fuzzy Logic Control* salah satu solusinya. Tujuan pembuatan alat tersebut adalah rancang bangun *hardware*, *software*, serta mengetahui unjuk kerja sistem penyiraman air otomatis dan proteksi hama tanaman menggunakan fuzzy logic control.

Proyek akhir ini dibuat berbasis ATMega328 yang diintegrasikan dengan sistem cerdas logika fuzzy menggunakan metode mamdani dan operator AND dalam fuzzyfikasi, serta *center of area* untuk menentukan defuzzyfikasinya. *Output* dari defuzzyfikasi sebagai *timer* untuk mengaktifkan relay. Minimum sistem berbasis ATMega328 sebagai kontroler utama sistem, Relay sebagai saklar pompa air, pestisida dan solenoid valve, Pompa air DC sebagai pemompa air dan pestisida, Solenoid valve sebagai katup keluarnya air, Sensor *soil moisture* dan suhu sebagai parameter utama, RTC sebagai pengatur waktu, serta LCD digunakan untuk menampilkan informasi.

Berdasarkan hasil pengujian yang dilakukan pada tanaman, sistem pada Penyiraman Air Otomatis dan Proteksi Hama Tanaman menggunakan *Fuzzy Logic Control* sudah berhasil. Terdapat *error* pada pembacaan sensor suhu sebesar 0,815 %. Dalam perancangan program fuzzy, nilai simulasi Matlab dengan algoritma diperoleh *error* 1,53 %. Dan untuk unjuk kerja keseluruhan pada tanaman secara keseluruhan telah sesuai dengan fungsi yang ditetapkan.

*Kata kunci* : Logika Fuzzy, ATMega328, DS18b20, RTC

## **MOTTO**

*“Maka sesungguhnya bersama kesulitan ada kemudahan, sesungguhnya bersama kesulitan ada kemudahan” (QS. Al-Insyirah 5-6)*

*“Saling berlakulah jujur dalam ilmu dan jangan saling merahasiakan. Sesungguhnya berkhianat dalam ilmu pengetahuan lebih berat hukumannya dari pada berkhianat dalam harta”*

*“Beajarlah kalian ilmu untuk ketentraman dan ketenangan serta rendah hatilah pada orang yang kamu belajar darinya”*

*“Doa tanpa usaha adalah bohong dan usaha tanpa doa itu sombong”*

## **PERSEMBAHAN**

*Proyek akhir ini saya persembahkan kepada :*

*Orang Tua atas segala doa, dukungan dan keikhlasan yang selalu diberikan kepada  
saya*

*Seluruh Dosen yang dengan ikhlas selalu membimbing dan mendidik saya*

*Sahabat Penghianat (Dhea,Daffa,Dodi,Maul, dkk) atas dukungan, saran, kritikan  
dan hiburan selama ini*

*Teman-teman Kelas B Teknik Elektronika 2015 atas support dan kebersamaannya  
selama ini*

*Tim KRTI Dreamagination yang menopang jerih payah dalam perjuangan untuk  
memajukan robotika UNY*

*Semua pihak yang telah membantu kelancaran penggeraan proyek akhir*

## **KATA PENGANTAR**

Puji dan syukur saya panjatkan kehadirat Allah SWT, yang telah melimpahkan rahmat dan hidayah-Nya sehingga dapat terselesaikan Proyek Akhir dengan judul

### **“RANCANG BANGUN PENYIRAMAN AIR OTOMATIS DAN PROTEKSI HAMA TANAMAN MENGGUNAKAN FUZZY LOGIC CONTROL “**

Proyek Akhir ini dapat terselesaikan tidak lepas dari berbagai pihak yang telah memberikan bantuan guna menunjang pengerjaan proyek akhir ini. Oleh karena itu, pada kesempatan ini saya ingin menyampaikan ucapan banyak terima kasih kepada:

1. Dr. Fatchul Arifin, M.T selaku Ketua Jurusan Pendidikan Teknik Elektronika dan Informatika, Fakultas Teknik, Universitas Negeri Yogyakarta dan juga selaku Dosen Pembimbing Proyek Akhir yang telah memberikan bimbingan dan arahan untuk penyelesaian proyek akhir ini.
2. Dr. Widarto, M.Pd selaku Dekan Fakultas Teknik Universitas Negeri Yogyakarta.
3. Dr. Sri Waluyanti, M.Pd selaku Ketua Program Studi Teknik Elektronika D3, Fakultas Teknik, Universitas Negeri Yogyakarta.
4. Seluruh Dosen Pengajar Jurusan Pendidikan Teknik Elektronika dan Informatika, Fakultas Teknik, Universitas Negeri Yogyakarta atas bekal ilmu yang diberikan kepada saya.
5. Kedua Orang tua dan adik yang selalu mendukung dan mendoakan dengan keikhlasannya untuk kelangsungan penyelesaian proyek akhir ini.
6. Teman-teman Teknik Elektronika kelas B 2015 yang telah memberikan bantuan sehingga pembuatan proyek akhir ini dapat terselesaikan.
7. Tim KRTI-Dreamagination UNY yang telah berjuang untuk pengabdian dan menjunjung nama Universitas dalam ajang Kontes Robot Terbang Indonesia.

8. Semua pihak yang tidak dapat saya sebutkan satu persatu yang telah membantu penggerjaan proyek akhir ini.

Saya menyadari bahwa laporan ini masih memiliki banyak kelemahan dan jauh dari kesempurnaan. Oleh karena itu saran dan kritik akan senantiasa diharapkan dengan maksud penyempurnaan penggerjaan proyek akhir ini.

Semoga proyek akhir ini dapat bermanfaat untuk kelangsungan proses belajar bagi mahasiswa Universitas Negeri Yogyakarta.

Yogyakarta,3 Juni 2018

Seto Aji Pratama

## DAFTAR ISI

LEMBAR PERSETUJUAN.....	Error! Bookmark not defined.
PROYEK AKHIR .....	Error! Bookmark not defined.
LEMBAR PENGESAHAN .....	Error! Bookmark not defined.
PROYEK AKHIR .....	Error! Bookmark not defined.
LEMBAR PERNYATAAN KEASLIAN .....	iii
ABSTRAK .....	v
MOTTO .....	vi
PERSEMAWAHAN .....	vii
KATA PENGANTAR .....	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR .....	xiii
DAFTAR TABEL.....	xv
DAFTAR LAMPIRAN .....	xvi
BAB I .....	1
PENDAHULUAN .....	1
A. LATAR BELAKANG.....	1
B. IDENTIFIKASI MASALAH.....	3
C. BATASAN MASALAH.....	4
D. RUMUSAN MASALAH.....	4
E. TUJUAN .....	5
F. MANFAAT .....	5
G. KEASLIAN GAGASAN .....	6
BAB II.....	9
PENDEKATAN PENYELESAIAN MASALAH .....	9
A. KEBUTUHAN AIR .....	9
B. TANAMAN HOLTIKULTURA .....	12
C. HAMA DAN PENYAKIT TANAMAN .....	13

D. PESTISIDA .....	14
E. ARDUINO .....	17
F. SENSOR KELEMBABAN TANAH.....	20
G. SENSOR SUHU DS18B20 .....	21
H. RELAY.....	23
I. SOLENOID VALVE .....	24
J. POMPA AIR DC .....	25
K. RTC (Real Time Clock) DS1302.....	26
L. LCD (Liquid Crystal Display) .....	28
M. FUZZY LOGIC CONTROLLER.....	29
1. Pengantar Logika Fuzzy .....	29
2. Himpunan Fuzzy .....	30
3. Fungsi Keanggotaan .....	34
4. Operator Dasar Zadeh untuk Operasi Himpunan Fuzzy .....	39
5. Sistem Inferensi Fuzzy .....	40
6. Fuzzy Reasoning .....	45
BAB III .....	50
KONSEP RANCANGAN.....	50
A. ANALYSIS .....	50
1. Identifikasi Kebutuhan .....	51
2. Analisis Kebutuhan .....	52
B. DESIGN .....	57
C. DEVELOPMENT.....	87
D. IMPLEMENTATION .....	88
1. <i>Hardware</i> .....	89
2. <i>Software</i> .....	91
E. EVALUATION.....	91
1. Uji Fungsional .....	92
2. Uji Kinerja.....	96
F. SPESIFIKASI ALAT .....	97

G. PENGOPERASIAN ALAT.....	98
BAB IV .....	99
PENGUJIAN DAN PEMBAHASAN.....	99
A. HASIL PENGUJIAN .....	99
1. Pengujian Fungsional .....	100
2. Pengujian Kinerja.....	107
B. PEMBAHASAN .....	114
1. Pembahasan Fungsional .....	114
2. Pembahasan Kinerja .....	116
BAB V.....	118
KESIMPULAN DAN SARAN.....	118
A. KESIMPULAN .....	118
B. KETERBATASAN ALAT.....	119
C. PENELITIAN LANJUTAN.....	120
DAFTAR PUSTAKA .....	122
LAMPIRAN.....	124

## DAFTAR GAMBAR

Gambar 1. Arduino Uno .....	19
Gambar 2. Pin Map Arduino Uno .....	19
Gambar 3. Pin Map Sensor Kelembaban Tanah .....	20
Gambar 4. Pin Map Sensor Suhu DS18B20 .....	21
Gambar 5. Konfigurasi DS18B20 dalam dua mode .....	22
Gambar 6. Simbol dan bentuk fisik relay .....	23
Gambar 7. Solenoid Valve 220 Volt .....	24
Gambar 8. Pompa Air DC 12 Volt .....	25
Gambar 9. Block Diagram <i>Real Time Clock</i> DS1302 .....	27
Gambar 10. Pin Map <i>Real Time Clock</i> DS1302 .....	27
Gambar 11. Modul LCD 20x4 Character.....	29
Gambar 12. Himpunan : MUDA, PAROBAYA, dan TUA .....	31
Gambar 13. Himpunan Fuzzy Untuk Variable Umur .....	32
Gambar 14. Representasi Linear Naik .....	35
Gambar 15. Kurva Segitiga .....	36
Gambar 16. Kurva Trapesium .....	37
Gambar 17. Daerah Bahu pada Variabel TEMPERATURE .....	38
Gambar 18. Proses Defuzzyifikasi .....	44
Gambar 19. Fuzzy Reasoning Single Rule Single Antecedent .....	46
Gambar 20. Fuzzy Reasoning Single Rule Multiple Antecedent .....	47
Gambar 21. Fuzzy Reasoning Multiple Rule Multiple Antecedent .....	48
Gambar 22. Desain Frame Alat .....	58
Gambar 23. Desain Penyiraman Air dan Pestisida .....	58
Gambar 24. Desain Panel Box .....	59
Gambar 25. Blok Diagram Sistem .....	60
Gambar 26. Shield Arduino Uno .....	61
Gambar 27. Power Supply .....	62
Gambar 28. Rangkaian Modul Step Down .....	63
Gambar 29. Rangkaian LCD 20x4.....	63

Gambar 30. Modul Real Time Clock.....	64
Gambar 31. Fungsi Keanggotaan <i>input</i> suhu .....	65
Gambar 32. Contoh kondisi fuzzyfikasi suhu.....	66
Gambar 33. Fungsi Keanggotaan <i>input</i> kelembaban tanah.....	67
Gambar 34. Contoh kondisi fuzzyfikasi kelembaban tanah .....	68
Gambar 35. Fungsi Keanggotaan Variabel Waktu .....	69
Gambar 36. Ilustrasi Hasil berdasarkan Fuzzy Rule.....	71
Gambar 37. Hasil Komposisi Fuzzy Rule.....	73
Gambar 38. Nilai Tengah dari Setiap Fungsi Keanggotaan Waktu .....	76
Gambar 39. Flowchart Algoritma Fuzzyfikasi Variabel Suhu.....	78
Gambar 40. Flowchart Program Fuzzyfikasi Variabel Suhu .....	79
Gambar 41. Flowchart Algoritma Fuzzyfikasi Variabel Kelem Tanah .....	80
Gambar 42. Flowchart Program Fuzzyfikasi Variabel Kelem Tanah.....	81
Gambar 43. Flowchart Algoritma Fuzzy Rule .....	82
Gambar 44. Flowchart Program Fuzzy Rule .....	83
Gambar 45. Flowchart Algoritma Defuzzyfikasi.....	84
Gambar 46. Flowchart Program Defuzzyfikasi .....	85
Gambar 47. Flowchart Algoritma Sistem Keseluruhan .....	86
Gambar 48. Frame Alat.....	89
Gambar 49. Penyiraman Air dan Pestisida .....	90
Gambar 50. Panel Box .....	90
Gambar 51. Layout PCB Shield Mikrokontroller .....	91
Gambar 52. <i>Membership Function</i> Kelembaban Tanah .....	109
Gambar 53. <i>Membership Function</i> Suhu .....	110
Gambar 54. Pembuktian Rule dengan Matlab .....	113

## DAFTAR TABEL

Tabel 1. Deskripsi Pin pada LCD .....	29
Tabel 2. Kebutuhan I/O ATMega .....	60
Tabel 3. Fuzzy Rule .....	70
Tabel 4. Data Luas dan Titik Tengah .....	76
Tabel 5. Rencana Pengujian Power Supply dan Step Down.....	92
Tabel 6. Rencana Pengujian Sensor Suhu .....	93
Tabel 7. Rencana Pengujian Sensor Kelembaban Tanah .....	94
Tabel 8. Rencana Pengujian Solenoid Valve .....	94
Tabel 9. Rencana Pengujian Pompa Air .....	95
Tabel 10. Rencana Pengujian Logika Fuzzy .....	96
Tabel 11. Rencana Pengujian Unjuk Kerja Penyiraman Air.....	97
Tabel 12. Rencana Pengujian Unjuk Kerja Penyiraman Pestisida .....	97
Tabel 13. Hasil Pengujian Power Supply dan Step Down.....	100
Tabel 14. Hasil Pengujian Sensor Suhu dengan Termometer .....	101
Tabel 15. Hasil Pengujian Sensor Kelembaban Tanah .....	102
Tabel 16. Hasil Pengujian Solenoid Valve .....	103
Tabel 17. Hasil Pengujian Pompa Air DC .....	104
Tabel 18. Hasil Pengujian Logika Fuzzy .....	104
Tabel 19. Perhitungan Fuzzy Rule .....	109
Tabel 20. Hasil Pengujian Unjuk Kerja Penyiraman Air .....	108
Tabel 21. Perhitungan Fuzzy Rule .....	112
Tabel 22. Hasil Pengujian Unjuk Kerja Penyiraman Pestisida .....	114

## **DAFTAR LAMPIRAN**

Lampiran 1. Skema Rangkaian Keseluruhan .....	125
Lampiran 2. Skema Rangkaian dan Layout PCB Shield .....	126
Lampiran 3. Source Program .....	127
Lampiran 4. Datasheet ATMega328.....	141
Lampiran 5. Datasheet DS18B20 .....	145
Lampiran 6. Datasheet <i>Real Time Clock</i> .....	148

## **BAB I**

### **PENDAHULUAN**

#### **A. LATAR BELAKANG**

Perkembangan teknologi dan ilmu pengetahuan dalam dunia *Intelligent Control System* telah mengalami kemajuan pesat, sehingga menjadikan sistem kendali cerdas sebagai bagian dari teknologi masa depan. Saat ini, perkembangan teknologi dan ilmu pengetahuan sistem kendali cerdas telah mampu meningkatkan kualitas maupun kuantitas di berbagai sektor seperti pendidikan, industri, pertahanan, medis, pertanian dan lain sebagainya. Sektor-sektor tersebut ada pada Negara Maju dan Negara Berkembang di dunia.

Salah satu Negara Berkembang adalah Indonesia yang merupakan Negara agraris dengan dianugerahi kekayaan alam yang melimpah ditambah posisi Indonesia yang dinilai strategis. Mulai dari sisi geografis, Indonesia terletak pada daerah tropis yang memiliki curah hujan yang tinggi sehingga banyak jenis tumbuhan yang dapat hidup dan tumbuh dengan cepat. Sektor pertanian merupakan sektor penting dalam perekonomian bangsa Indonesia sehingga pemerintahan aktif meningkatkan produktifitas disektor pertanian. Lewat rilisan data Badan Pusat Statistik (BPS) pada Triwulan II 2017 menunjukan bahwa sektor pertanian memberi kontribusi positif untuk pertumbuhan ekonomi Indonesia. Berdasarkan data itu, terlihat bahwa besaran Produk Domestik Bruto (PDB) mencapai Rp. 3.366,8 triliun. Kemudian, dilihat dari sisi produksi, sektor pertanian

merupakan sektor kedua yang paling berpengaruh terhadap pertumbuhan ekonomi setelah industry pengolahan dan posisinya masih diatas sektor perdangan dan konstruksi. Triwulan II 2017 ini, sektor pertanian dalam arti luas menyumbang sebanyak 13,92 persen, sementara pada Triwulan I 2017 kontribusinya 13,59 persen. Pada konferensi pers di Kantor BPS , Kepala BPS Suhariyanto menyampaikan bahwa Indonesia berada 6,9 persen di atas China dengan hasil yang cukup bagus.

Pemerintahan Kabupaten Sleman Provinsi Daerah Istimewa Yogyakarta ikut berpartisipasi dalam meningkatkan hasil di sektor pertanian dengan menjalankan program-program kerja bidang ini. Salah satu programnya yaitu Gerakan Pemanfaatan Pekarangan dalam Rangka menuju kemandirian pangan. Program ini diharapkan dapat memaksimalkan pemanfaatan pekarangan rumah untuk bercocok tanam sehingga menghasilkan tanaman holtikultura yang dapat dikonsumsi sendiri maupun dijual belikan.

Pelaksanaan program terdapat kendala yang menyebabkan kerugian signifikan, seperti faktor kelalaian manusia (Waktu penyiraman dan Lama penyiraman), faktor gangguan organisme kecil (virus, bakteri, jamur) atau hama tanaman yang disampaikan oleh salah satu petugas Dinas Pertanian, Pangan, dan Perikanan Kabupaten Sleman Indah Lestari dalam wawancaranya bahwa masyarakat yang menjalankan program ini masih menggunakan metode manual sehingga kurang konsisten dalam melakukan penyiraman air dan peptisida pada tanaman dalam pot secara teratur. Selain itu penggunaan pestisida yang tidak

sesuai dengan prosedur akan menyebabkan hama yang lebih kebal. Beberapa diantaranya membutuhkan penanganan yang lebih serius. Pengendalian gangguan organisme kecil atau hama tanaman dan takaran pestisida yang akan digunakan merupakan konsep yang harus diterapkan secara terpadu dan berkelanjutan. Adapun penggunaan alat penyiraman saat ini masih dilakukan secara manual oleh manusia.

Berdasarkan permasalahan di atas tersebut maka penulis mendapat ide untuk membuat alat yang berfungsi sebagai penyiraman air otomatis dan proteksi hama pada tanaman yang dapat menyiram air sesuai kebutuhan dan melindungi tanaman dari gangguan organisme kecil serta dapat bekerja secara otomatis dengan menggunakan sistem kendali cerdas. Keberadaan alat ini diharapakan mampu membantu masyarakat dalam melakukan kegiatan bercocok tanam sehingga mendapatkan hasil yang lebih baik ataupun mempunyai nilai jual tinggi.

## **B. IDENTIFIKASI MASALAH**

Berdasarkan paparan dari latarbelakang masalah, dapat diidentifikasi masalah sebagai berikut:

1. Manusia sering menyiram tanaman holtikultura tanpa memperkirakan waktu dan lamanya penyiraman.
2. Gangguan organisme kecil (virus, bakteri, jamur) atau hama tanaman yang dapat merugikan tanaman holtikutura.

3. Penyiraman tanaman holtikultura masih menggunakan metode manual tanpa menggunakan sistem kendali cerdas.

### **C. BATASAN MASALAH**

Berdasarkan identifikasi masalah yang disebutkan, perlu adanya pembatasan masalah sehingga ruang lingkup permasalahan jelas. Ruang lingkup batasan masalah dalam proyek ini adalah penyiraman tanaman holtikultura masih menggunakan metode manual tanpa menggunakan sistem kendali cerdas dan gangguan organisme kecil (virus, bakteri, jamur) atau hama tanaman yang dapat merugikan tanaman holtikultura.

### **D. RUMUSAN MASALAH**

Berdasarkan identifikasi dan batasan masalah diatas, maka dapat dibuat rumusan sebagai berikut:

1. Bagaimana Rancang Bangun Perangkat Keras (*Hardware*) Penyiraman Air Otomatis dan Proteksi Hama Tanaman menggunakan *Fuzzy Logic Control*?
2. Bagaimana Rancang Bangun Perangkat Lunak (*Software*) untuk Sistem Penyiraman Air Otomatis dan Proteksi Hama Tanaman menggunakan *Fuzzy Logic Control*?
3. Bagaimana unjuk kerja Sistem Penyiraman Air Otomatis dan Proteksi Hama Tanaman menggunakan *Fuzzy Logic Control*?

## **E. TUJUAN**

Dengan merujuk pada perumusan masalah yang telah dijabarkan, maka diharapkan dapat mencapai tujuan sebagai berikut:

1. Rancang bangun perangkat keras (*Hardware*) Penyiraman Air Otomatis dan Proteksi Hama Tanaman menggunakan *Fuzzy Logic Control*.
2. Rancang bangun perangkat lunak (*Software*) untuk Sistem Penyiraman Air Otomatis dan Proteksi Hama Tanaman menggunakan *Fuzzy Logic Control*.
3. Mengetahui unjuk kerja Sistem Penyiraman Air Otomatis dan Proteksi Hama Tanaman menggunakan *Fuzzy Logic Control*.

## **F. MANFAAT**

Manfaat yang dapat diambil dari pembuatan proyek akhir ini adalah sebagai berikut:

1. Bagi Mahasiswa
  - a. Sebagai sumber rujukan untuk pengembangan ilmu tentang sistem kendali cerdas dengan logika fuzzy.
  - b. Sebagai sarana untuk mengaplikasikan teori dan praktikum sistem kendali cerdas yang didapatkan dalam perkuliahan.
  - c. Sebagai wujud kontribusi terhadap Universitas baik dalam citra maupun daya tawar terhadap masyarakat luas.

- d. Memberikan motivasi kepada mahasiswa untuk tetap berkarya dan menjadi salah satu pelaku dalam kemajuan teknologi
- 2. Bagi Perguruan Tinggi
  - a. Terciptanya alat yang inovatif dan bermanfaat sebagai sarana ilmu pengetahuan dan media pembelajaran.
  - b. Sebagai referensi tambahan untuk mengaplikasikan ilmu terapan sistem kendali cerdas dengan logika fuzzy.
  - c. Sebagai tolak ukur daya serap mahasiswa yang bersangkutan selama menempuh pendidikan dan kemampuan ilmunya secara praktis.
- 3. Bagi Masyarakat, Peneliti, dan Dunia Pertanian
  - a. Memudahkan proses penyiraman tanaman secara otomatis pada pertanian dan perkebunan dengan menggunakan sistem kendali cerdas.
  - b. Meningkatkan kredibilitas produksi dari pertanian dan perkebunan,
  - c. Sebagai sarana rujukan penelitian dibidang pertanian dan perkebunan.

## **G. KEASLIAN GAGASAN**

Proyek akhir dengan judul “ Rancang Bangun Penyiraman Air Otomatis dan Proteksi Hama Tanaman menggunakan *Fuzzy Logic Control* “ merupakan proyek akhir yang didesain maksud dan tujuan sebagai inovasi baru dibidang pertanian dan perkebunan serta pengaplikasian sistem kendali cerdas menggunakan fuzzy logic control. Adapun karya sebagai rujukan pembuatan proyek akhir adalah *Alat*

*Penyiraman Tanaman Otomatis dengan Logika Fuzzy Berbasis Atmega 16* yang di kerjakan oleh Bayu Aji Kurniawan sebagai Proyek Akhir tahun 2015 dari program studi Teknik Elektronika, Pendidikan Teknik Elektronika dan Informatika Universitas Negeri Yogyakarta. Implementasi *Alat Penyiraman Tanaman Otomatis dengan Logika Fuzzy Berbasis Atmega 16* yang dikerjakan oleh Bayu Aji Kurniawan berfokus pada suhu dan kelembaban tanah sebagai variabel *input*. Penggunaan software GUI Fuzzy dari Matlab sebagai fuzzifikasi dengan metode mamdani. Driver dan Motor DC sebagai variabel *output*, kemudian dibuat rule yang sekiranya sesuai dengan kebutuhan penyiraman. Dan Liquid Crystal Display (LCD) untuk menampilkan nilai suhu dan nilai kelembaban tanah.

Adapun perbedaan Alat yang dibuat pada proyek akhir ini dengan proyek akhir yang dibuat Bayu Aji Kurniawan adalah:

1. Karya Bayu Aji Kurniawan menggunakan motor DC dan driver motor sedangkan penulis menggunakan pompa air 12 volt tanpa driver pada sistem penyiraman air dan pestisida.
2. Karya Bayu Aji Kurniawan menggunakan sensor suhu LM35 sedangkan penulis menggunakan sensor suhu DS18b20 yang mempunyai akurasi lebih tinggi.
3. Karya Bayu Aji Kurniawan menggunakan mikrokontroller ATmega 16 sedangkan penulis menggunakan ATMega 328 yang mempunyai PORT I/O lebih banyak

*Penerapan Logika Fuzzy pada Sistem Penyiraman Tanaman Otomatis Berbasis Mikrokontroller* yang dikerjakan oleh Tulus Pranata, Beni Irawan, Ilhamsyah dalam Penelitian Jurusan Sistem Komputer, Fakultas MIPA Universitas Tanjungpura. Pada penelitian yang dikerjakan oleh Tulus Pranta, Beni Irawan, dan Ilhamsyah, menggunakan Real Time Clock (RTC) DS1307 sebagai penyimpan waktu dan tanggal. Adapun pada penelitian tersebut fokus kepada tanaman seledri (*Apium gravolens L*) sebagai objek utama penelitian.

Perbedaan proyek akhir ini dengan yang dibuat oleh Tulus Pranata, Beni Irawan, dan Ilhamsyah adalah:

1. Karya Tulus Pranata dkk menggunakan sistem logika fuzzy dengan metode sugeno sedangkan penulis menggunakan metode mamdani yang mempunyai sifat *human inferensi system*, dimana metode yang digunakan sesuai dengan aturan-aturan pada manusia umumnya.
2. Karya Tulus Pranata dkk menggunakan sensor suhu LM35 sedangkan penulis menggunakan sensor suhu DS18b20.

## **BAB II**

### **PENDEKATAN PENYELESAIAN MASALAH**

#### **A. KEBUTUHAN AIR**

Air merupakan salah satu komponen fisik yang vital dan diperlukan dalam jumlah banyak untuk pertumbuhan dan perkembangan tanaman. Air juga berfungsi sebagai stabilisator suhu tanaman (Suhartono,2008). Sekitar 85-90% dari bobot segar sel dan jaringan tanaman tinggi ada pada air, adapun definisi tentang kebutuhan air suatu tanaman adalah sebagai berikut “Jumlah air yang diperlukan untuk memenuhi kehilangan air melalui evapotranspirasi (ET-tanaman) tanaman yang sehat, tumbuh pada sebidang lahan yang luas dengan kondisi tanah yang tidak mempunyai kendala (kendala lengas tanah dan kesuburan tanah) dan mencapai potensi produksi penuh pada kondisi lingkungan tumbuh tertentu” untuk menghitung ET-tanaman direkomendasikan suatu prosedur 3 tahap yaitu:

1. Pengaruh iklim terhadap kebutuhan air tanaman diberikan oleh ETo (Evapotranspirasi tanaman referensi), yaitu “laju evapotranspirasi dari permukaan berumput luas setinggi 8-15 cm, rumput hijau yang tingginya seram, tumbuh aktif, secara lengkap menaungi permukaan tanah dan tidak kekurangan air”. Empat metode yang digunakan adalah Blaney-Criddle, Radiasi, Penman dan Evaporasi panic, dimodifikasi untuk menghitung ETo dengan menggunakan data iklim harian selama periode 10 atau 30 hari.

2. Pengaruh karakteristik tanaman terhadap kebutuhan air tanaman diberikan oleh koefisien tanaman ( $kc$ ) yang menyatakan hubungan antara  $ETo$  dan  $ET$  tanaman ( $ET_{tanaman} = kc \cdot ETo$ ). Nilai-nilai  $kc$  beragam dengan jenis tanaman, fase pertumbuhan tanaman, musim pertumbuhan, dan kondisi cuaca yang ada.
3. Pengaruh kondisi local dan praktik pertanian terhadap kebutuhan air tanaman, termasuk variasi local cuaca, tinggi tempat, ukuran petak lahan, adveksi angin, ketersedian lengas lahan, salinitas, metode irigasi dan kultivitas tanaman.

Beberapa pendekatan dapat digunakan untuk perencanaan pemanfaatan sumberdaya air secara optimal dalam sistem produksi pertanian. Informasi pokok yang diperlukan adalah mengenai sumberdaya air, lahan, dan tanaman. Khusus dalam kaitannya dengan pekarangan, maka informasi yang diperlukan adalah sumberdaya air (air hujan, air tanah, dan air irigasi permukaan), sifat dari ciri tanah, dan syarat tumbuh berbagai tanaman pekarangan. Berdasarkan atas informasi ini maka dapat disusun alternative sistem produksi pada lahan pekarangan. Beberapa parameter penting adalah:

- a. Pemilihan tanaman, beberapa faktor yang juga harus dipertimbangkan adalah jumlah air yang tersedia, kondisi tanah dan iklim, preferensi petani, kebutuhan tenaga kerja dan modal, peluang pasar dan tingkat teknologi. Penyusunan pola tanam dilakukan sesuai dengan neraca lengas lahan.
- b. Intensitas pertanaman (*Cropping Intensity*), intensitas ini bervariasi antar waktu (musim) dan lokasi lahan. Hal ini berkaitan erat dengan tingkat investasi.

- c. Tingkat penyediaan air irigasi ditentukan oleh air irigasi, neraca lengas lahan, pola tanam dan intensitas pertanaman.
- d. Metode irigasi, Pemilihan metode irigasi harus dilakukan pada awal periode perencanaan. Pertimbangannya meliput investasi, efisiensi penggunaan air, kemudahan penerapan, dan kesesuaian dengan kondisi local, erodibilitas tanah, laju infiltrasi, salinitas air dan lainnya.
- e. Drainase dan pencucian, Drainase yang baik diperlukan untuk menunjang keberhasilan program irigasi lahan pekarangan. Untuk menghindari akumulasi garam pada zone perakaran tanaman dan kemungkinan kerusakan tanaman yang diakibatkannya, maka kebutuhan pencucian harus ditentukan secara tepat.

Kekurangan air pada jaringan tanaman dapat menurunkan turgor sel, meningkatkan konsentrasi makro molekul serta mempengaruhi membrane sel dan potensi aktivitas kimia air dalam tanaman (Mubiyanto,1997). Mengingat pentingnya peran air, waktu penyiraman juga harus diperhatikan karena waktu yang baik untuk melakukan penyiraman adalah pada saat pagi dan sore hari. Dan tidak disarankan menyiram siang hari karena Sel-sel pada tanaman akan mengkerut disiram dalam keadaan panas, sel-sel tersebut akan menggelembung terisi air lalu akan pecah. Jika hal ini terus terulang, maka lama kelamaan tanaman akan rusak dan mati (Tulus Pranata,2015). Harwono (1993) berpendapat bahwa cekaman kekurangan air dapat menghambat aktivitas fotosintesis dan distribusi

asimilat ke dalam organ reproduktif. Pemberian air yang berbeda akan menimbulkan respon tanaman yang berbeda pula.

## **B. TANAMAN HOLTIKULTURA**

Holtikultura berasal dari bahasa latin, yaitu *hortur* (kebun) dan *colere* (menumbuhkan) . Menurut Perkumpulan Holtikultura Indonesia (2004) adalah gabungan ilmu, seni, dan teknologi dalam mengolala tanaman sayuran, buah, ornament, bumbu-bumbu dan tanaman obat-obatan. Tanaman holtikultura mencakup lebih luas yakni budidaya tanaman sayuran, buah-buahan dan berbagai macam tanaman hias. Tanaman holtikultura memiliki prospek pengembangan yang baik karena memiliki nilai ekonomi yang tinggi dan potensi pasar yang terbuka lebar baik di dalam negeri maupun di luar negeri.

Pada tahun 2008 menurut Direktorat Jendral Holtikultura, yang termasuk komoditas unggulan nasional holtikultura berupa duku, semangka, nanas, salak, melon, sirsak, suran organik, sayuran dataran rendah dan tanaman hias. Berdasarkan kegunaannya, tanaman holtikultura dapat dikelompokan menjadi tanaman holtikultura yang dikonsumsi, yakni sayuran, buah-buahan, dan tanaman holtikultura yang tidak dikonsumsi, yaitu tanaman hias.

### **C. HAMA DAN PENYAKIT TANAMAN**

“Yang dimaksud dengan hama ialah semua binatang yang mengganggu dan merugikan tanaman yang diusahakan manusia” (Pracaya, 2003:5). Bagian tanaman yang diganggu tidak hanya satu bagian melainkan dapat seluruh bagian. Kerugian yang ditimbulkan oleh hama mempunyai kisaran yang besar, dari tidak berarti sampai yang dapat menggagalkan panen. Besar atau banyaknya kerugian yang diderita dipengaruhi oleh beberapa jenis hamanya, salah satun hama adalah nematode, tikus, gulma atau bentuk lain dari kehidupan terestrial atau air yang berbahaya bagi kesehatan manusia atau ternak, serta mengganggu kegiatan ekonomi seperti pertanian atau perkebunan.

“Tanaman dikatakan sakit bila ada perubahan seluruh atau sebagian organ-organ tanaman yang menyebabkan terganggunya kegiatan fisiologis sehari-hari. Secara singkat penyakit tanaman adalah penyimpangan dari keadaan normal” (Pracaya, 2003:320). Tanaman dapat dikatakan sehat atau normal jika tanaman tersebut dapat menjalankan fungsi-fungsi fisiologi dengan baik dan benar, seperti pembelahan sel dan perkembangan sel, penghisapan air dan unsur zat hara, fotosintesis dan lain-lain. Gangguan pada fungsi-fungsi tanaman dapat menimbulkan penyakit.

Pada tanaman sayuran juga tidak lepas dari serangan hama dan penyakit tanaman yang menyebabkan kerugian pada tanaman sayuran. Maka dari itu perlu adanya pencegahan dan pengamatan hama tanaman pada tanaman sayuran karena

jika sampai terlambat serangan penyakit dan hama akan sulit untuk dikendalikan.

Adapun beberapa jenis hama yang menyerang tanaman adalah sebagai berikut:

1. Hama Ulat Grayak (*Spodoptera Litura Fabricus*)
2. Hama Tungau Merah (*Tetranychus SP.*) dan Tungau Kuning (*Polyphagotarsonemus Latus*)
3. Hama Kutu Daun Persik (*Myzus Persicae*)
4. Hama Kutu Daun (*Aphididae*)
5. Hama Kutu Kebul (*Bemisi Tabachi*)
6. Hama Trhips (Trhips Parvispinus Karny)
7. Hama Lalat Buah (*Bactrocera SP*)
8. Hama Ulat Tanah (*Helicoverpa SP* dan *Spodoptera Exigua*)

#### **D. PESTISIDA**

Pestisida adalah substansi kimia dan bahan lain serta jasad renik dan virus yang digunakan untuk mengendalikan berbagai hama pengganggu tumbuhan.

Menurut PP No.7 Tahun 1973, yang dimaksud pestisida adalah semua zat kimia dan bahan lain serta jasad renik dan virus yang dipergunakan untuk:

1. Memberantas atau mencegah hama-hama dan penyakit yang merusak tanaman, bagian-bagian tanaman atau hasil-hasil tanaman.
2. Memberantas rerumputan atau tanaman pengganggu/gulma.
3. Mematikan daun dan mencegah pertumbuhan yang tidak diinginkan.

4. Mengatur atau merangsang pertumbuhan tanaman atau bagian-bagian tanaman, tidak termasuk pupuk.
5. Memberantas atau mencegah hama-hama ulat pada hewan-hewan peliharaan dan ternak.
6. Memberantas atau mencegah hama-hama air.
7. Memberantas atau mencegah binatang-binatang dan jasad-jasad renik dalam rumah tangga, bangunan dan alat-alat pengangkutan.
8. Memberantas atau mencegah binatang-binatang yang dapat menyebabkan penyakit pada manusia dan binatang yang perlu dilindungi dengan penggunaan pada tanaman, tanah dan air.

Dalam UU No.12 Tahun 1992 tentang sistem budidaya tanaman. Pestisida merupakan zat pengatur dan perangsang tumbuh bahan lain serta organisme renik atau virus yang digunakan untuk melakukan perlindungan tanaman. Pestisida merupakan bahan yang telah banyak memberikan manfaat untuk keberlangsungan dunia produksi pertanian. Banyaknya Organisme Pengganggu Tumbuhan (OPT) yang dapat menurunkan hasil panen dapat diminimalisir dengan pestisida, sehingga kehilangan hasil akibat OPT tidak terlalu besar. Selain bidang pertanian pestisida juga memberikan banyak manfaat untuk membantu masalah yang timbul akibat adanya organisme pengganggu di tingkat rumah tangga, seperti pembasmian nyamuk. Dengan adanya pestisida maka proses pembasmian nyamuk akan menjadi lebih cepat dan efisien. Bahkan masih banyak lagi peranan pestisida bagi kehidupan diberbagai bidang.

Pestisida pada tanaman sayuran mempunyai peranan penting bagi kehidupan tanaman itu sendiri. Dengan adanya pestisida, hama dan penyakit tanaman yang menyerang tanaman sayuran akan musnah dan hasil panen sayuran lebih baik. Penggunaan pestisida pada tanaman sayuran disesuaikan dengan hama dan penyakit yang menyerang tanaman sayuran sehingga lebih efisien dan efektif. Berikut macam pestisida yang dapat digunakan adalah sebagai berikut:

1. Fungisida merupakan jenis pestisida yang digunakan untuk mengendalikan penyakit yang disebabkan oleh jamur atau cendawan.
2. Akarisida merupakan jenis pestisida yang digunakan untuk mengendalikan hama jenis kutu-kutuan pada tanaman sayuran.
3. Insektisida merupakan jenis pestisida yang digunakan untuk mengendalikan hama serangga pada tanaman sayuran.
4. Bakterisida merupakan jenis pestisida yang digunakan untuk mengendalikan penyakit pada tanaman sayuran yang disebabkan oleh bakteria.
5. Nematisida merupakan jenis pestisida yang digunakan untuk mengendalikan hama didalam tanah, seperti nematode/cacing atau puru akar.
6. Pupuk Daun / Pupuk Cair digunakan untuk mengendalikan hama maupun penyakit, namun pupuk daun juga termasuk pestisida.
7. ZPT (Zat Pengatur Tumbuh) juga bukan material untuk membunuh hama maupun penyakit, akan tetapi ZPT termasuk juga sebagai pestisida.

## **E. ARDUINO**

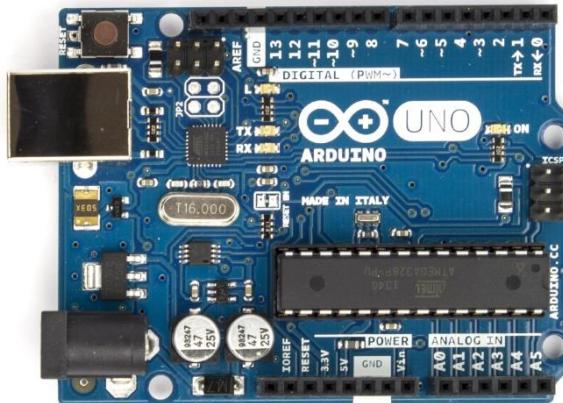
Arduino adalah kit elektronik atau papan elektronik *open source* yang didalamnya terdapat komponen utama yaitu sebuah chip mikrokontroller dengan jenis AVR dari perusahaan Atmel (<http://it-jurnal.com>). Sedangkan mikrokontroller sendiri itu adalah chip atau IC (*integrated circuit*) yang bisa diprogram menggunakan komputer. Tujuan memberikan program pada mikrokontroller adalah agar rangkaian dapat membaca *input*, memproses input tersebut kemudian menghasilkan *output* sesuai yang diinginkan. Arduino menggunakan keluarga mikrokontroller ATMega yang dirilis oleh Atmel sebagai basis, namun terdapat individu/perusahaan yang membuat *clone* arduino dengan menggunakan mikrokontroller lain dan tetap kompatibel dengan arduino pada level *hardware*. Agar dapat fleksibel, program dimasukan melalui bootloader meskipun ada opsi untuk membypass bootloader dan menggunakan downloader untuk memprogram mikrokontroller secara langsung melalui port ISP. Seperti halnya mikrokontroller yang banyak jenisnya, Arduino lahir dan berkembang kemudian muncul dengan berbagai jenis, diantaranya adalah sebagai berikut:

### **1. Arduino Uno**

Arduino UNO adalah board mikrokonroller berbasis ATmega328 berfungsi sebagai pengontrol rangkaian elektronik dan umunya dapat menyimpan program didalamnya serta *open source*. Pada Arduino Uno terdapat bagian penting seperti IC (*Integrated Circuit*), memori, I/O, dan unit pendukung seperti Analog to Digital

Converter (ADC) yang sudah terintegrasi didalamnya seperti terlihat pada Gambar 1 dan Gambar 2. Berikut ini adalah spesifikasi dari Arduino Uno antara lain:

- a. Chip mikrokontroller : ATmega328
- b. Tegangan Operasi : 5V
- c. Tegangan *input* (rekомendasi) : 7-12V
- d. Tegangan *input* (limit) : 6-20V
- e. Digital I/O pin : 14 pin digital (6 diantaranya PWM)
- f. Analog input pin : 6 pin
- g. Arus DC per pin I/O : 40mA
- h. Arus DC pin 3.3V : 50mA
- i. Memory flash : 32 KB (ATmega328)
- j. SRAM : 2 KB (ATmega328)
- k. EEPROM : 1 KB (ATMega328)
- l. Clock speed : 16 MHz
- m. Dimensi : 68.6 mm x 53.4 mm
- n. Berat : 25g



**Gambar 1. Arduino Uno**

Sumber: [www.arduino.cc](http://www.arduino.cc)

### Atmega168 Pin Mapping

Arduino function			Arduino function		
reset	(PCINT14/RESET)	PC6	1 28	PC5 (ADC5/SCL/PCINT13)	analog input 5
digital pin 0 (RX)	(PCINT16/RXD)	PD0	2 27	PC4 (ADC4/SDA/PCINT12)	analog input 4
digital pin 1 (TX)	(PCINT17/TXD)	PD1	3 26	PC3 (ADC3/PCINT11)	analog input 3
digital pin 2	(PCINT18/INT0)	PD2	4 25	PC2 (ADC2/PCINT10)	analog input 2
digital pin 3 (PWM)	(PCINT19/OC2B/INT1)	PD3	5 24	PC1 (ADC1/PCINT9)	analog input 1
digital pin 4	(PCINT20/XCK/T0)	PD4	6 23	PC0 (ADC0/PCINT8)	analog input 0
VCC		VCC	7 22	GND	GND
GND		GND	8 21	AREF	analog reference
crystal	(PCINT6/XTAL1/TOSC1)	PB6	9 20	AVCC	VCC
crystal	(PCINT7/XTAL2/TOSC2)	PB7	10 19	PB5 (SCK/PCINT5)	digital pin 13
digital pin 5 (PWM)	(PCINT21/OC0B/T1)	PD5	11 18	PB4 (MISO/PCINT4)	digital pin 12
digital pin 6 (PWM)	(PCINT22/OC0A/AIN0)	PD6	12 17	PB3 (MOSI/OC2A/PCINT3)	digital pin 11(PWM)
digital pin 7	(PCINT23/AIN1)	PD7	13 16	PB2 (SS/OC1B/PCINT2)	digital pin 10 (PWM)
digital pin 8	(PCINT0/CLK0/ICP1)	PB0	14 15	PB1 (OC1A/PCINT1)	digital pin 9 (PWM)

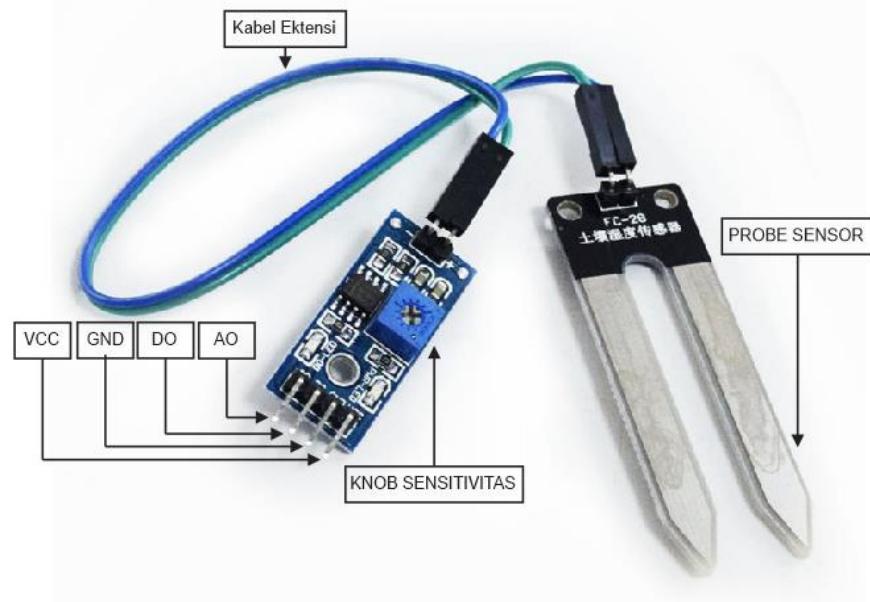
Digital Pins 11,12 & 13 are used by the ICSP header for MOSI, MISO, SCK connections (Atmega168 pins 17,18 & 19). Avoid low-impedance loads on these pins when using the ICSP header.

**Gambar 2. Pin Map ATMega/168/328/Arduino Uno**

Sumber : [www.Arduino.cc](http://www.Arduino.cc)

## F. SENSOR KELEMBABAN TANAH

Sensor kelembaban tanah merupakan sensor yang terdiri dari kombinasi modul pengkondisian sinyal (YL-39) dan probe sensor (YL-69). Probe sensor terdiri dari dua probe yang berfungsi untuk melewatkkan arus melalui tanah, kemudian membaca resistansinya untuk mendapatkan nilai tingkat kelembaban. Berikut ini adalah pin map sensor kelembaban tanah ditunjukan pada gambar 3.



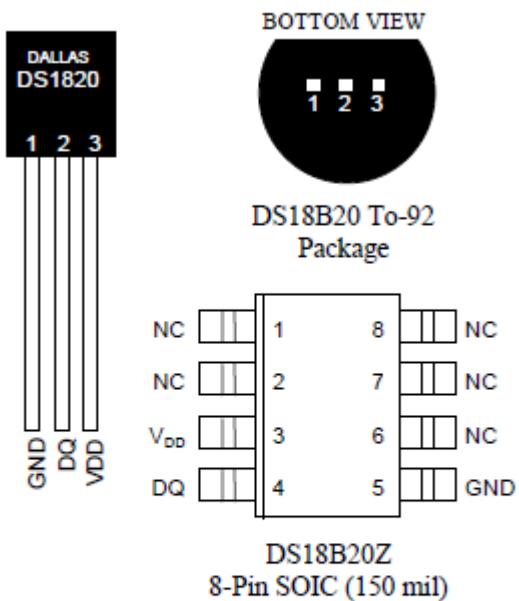
**Gambar 3. Pin Map Sensor Kelembaban Tanah**

Modul sensor kelembaban tanah memiliki 4 pin, yaitu GND (ground), VCC (3.3 – 5 volt), AO (keluaran analog yang akan dibaca oleh Arduino), dan DO (dapat diatur sensitivitasnya menggunakan knobe pengatur, dan menghasilkan logika digital HIGH/LOW pada level kelembaban tertentu. Pada proyek akhir ini hanya tiga pin yang dimanfaatkan, yaitu GND, VCC dan AO.

## G. SENSOR SUHU DS18B20

Sensor DS18B20 merupakan sensor digital yang memiliki 12-bit ADC internal. Sensor suhu DS18B20 memiliki presisi yang tinggi, sebab jika tegangan referensii sebesar 5 volt, maka akibat perubahan suhu terkecil sebesar  $5/(2^{12}-1) = 0.0012$  Volt. Pada rentang suhu -10 sampai +85 derajat celcius, sensor ini memiliki akurasi  $+/-0.5$  derajat. Sensor ini bekerja menggunakan protocol komunikasi 1-wire (one-wire).

### PIN ASSIGNMENT



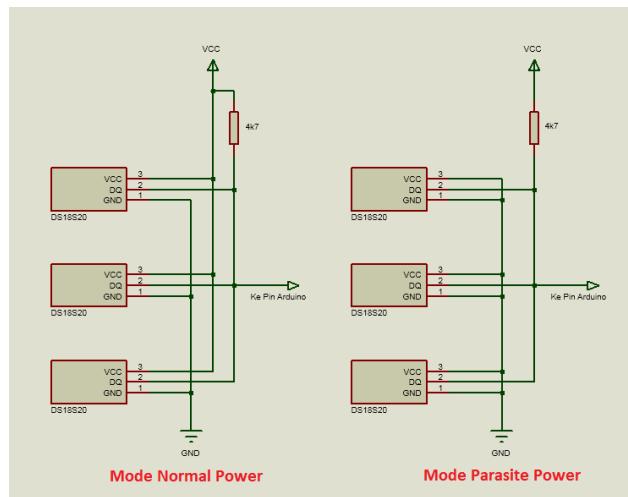
### PIN DESCRIPTION

- GND - Ground
- DQ - Data In/Out
- V<sub>DD</sub> - Power Supply Voltage
- NC - No Connect

**Gambar 4. Pin Map Sensor Suhu DS18B20**

Sumber : Datasheet DS18b20

Pada gambar 4 IC DS18B20 memiliki tiga kaki, yaitu GND (ground, pin 1), DQ (Data, pin 2), VDD (power, pin 3). Sensor dapat bekerja dalam dua mode, yaitu mode **normal power** dan mode **parasite power**. Pada **Mode Normal**, GND akan terhubung dengan ground, VCC akan terhubung dengan 5v dan DQ akan terhubung dengan pin arduino, namun ditambahkan resistor pull-up 4,7k. Mode ini direkomendasikan pada aplikasi yang melibatkan banyak sensor dan membutuhkan jarak panjang. Sedangkan **Mode Parasite** , GND dan VCC disatukan dan terhubung dengan ground. DQ akan terhubung dengan pin arduino melalui resistor pull-up. Pada mode ini, power diperoleh dari power data. Mode ini bisa digunakan untuk aplikasi yang melibatkan sedikit sensor dalam jarak yang pendek. Berikut ini konfigurasi DS18B20 dalam dua mode dapat dilihat pada gambar 5



**Gambar 5. Konfigurasi DS18B20 dalam dua mode**

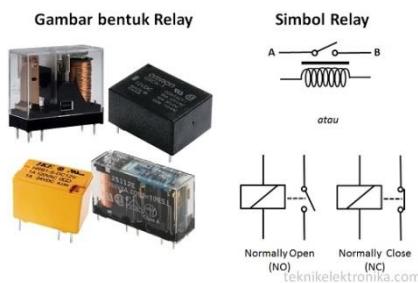
Sumber : [www.tutorkeren.com](http://www.tutorkeren.com)

## H. RELAY

Relay adalah saklar (*switch*) elektrik yang bekerja berdasarkan medan magnet. Relay terdiri dari suatu lilitan dan *switch* mekanik. *Switch* mekanik akan bergerak jika ada arus listrik yang mengalir melalui lilitan yang terdapat pada relay ([http://p\\_musa.staff.gunadarma.ac.id](http://p_musa.staff.gunadarma.ac.id)). Susunan kontak pada relay dapat dilihat pada gambar 6 adalah sebagai berikut:

*Normally Open* : relay akan menutup bila dialiri arus listrik

*Normally Close* : relay akan membuka bila dialiri arus listrik



**Gambar 6. Simbol dan bentuk fisik relay**

Sumber: <http://teknikelektronika.com/pengertian-relay-fungsi-relay/>

Pada proyek akhir ini relay digunakan sebagai saklar untuk mengaktifkan dan menonaktifkan pompa air dc dan solenoid valve. Relay tersebut dikontrol oleh logika mikrokontroller dengan bantuan transistor NPN yang berfungsi sebagai penguat arus. Adapun karakteristik relay yang digunakan adalah sebagai berikut:

1. Tegangan kerja DC (5V).
2. Prinsip kerja mekanik : Engsel.
3. Cocok untuk beban kecil.
4. Kontak relative kecil.

## I. SOLENOID VALVE

Solenoid valve adalah suatu katup yang digunakan oleh energi listrik melalui solenoid, mempunyai kumparan sebagai penggeraknya yang dapat menggerakan piston. Solenoid valve dapat digerakkan menggunakan arus DC ataupun AC. Tugas dari solenoid valve adalah untuk *shut-off, release, dose, distribute ataupun mix fluids*.

Prinsip kerja dari solenoid valve yaitu katup listrik yang mempunyai koil sebagai penggeraknya dimana ketika koil mendapat supply tegangan maka koil tersebut akan berubah menjadi medan magnet sehingga menggerakan piston pada bagian dalamnya, ketika piston bertekanan yang berasal dari supply (*service uniy*). Adapun pada gambar 7 menggunakan solenoid valve untuk fluida dengan tegangan kerja 220 Volt.



**Gambar 7. Solenoid Valve 220 Volt**

## J. POMPA AIR DC

Pompa air atau *Water pump* adalah elemen yang berfungsi untuk menyerap sekaligus mendorong air, dengan cara memindahkan sejumlah volume air melalui ruang suction menuju ke ruang outlet dengan menggunakan impeler, sehingga seluruh ruang udara terisi oleh air dan menimbulkan tekanan fluida untuk ditarik melalui dasar penampungan menuju keluar. Air yang terdapat pada impeler akan digerakan menggunakan sebuah motor. Selama impeler tersebut berputar, air akan terus didorong keluar menuju ke pipa penyaluran atau outler air. Untuk menjalankan pompa dapat digunakan tegangan kerja AC ataupun DC.



**Gambar 8. Pompa Air DC 12 Volt**

Pada gambar 8 merupakan pompa air DC, adapun karakteristik pompa air DC yang digunakan adalah sebagai berikut:

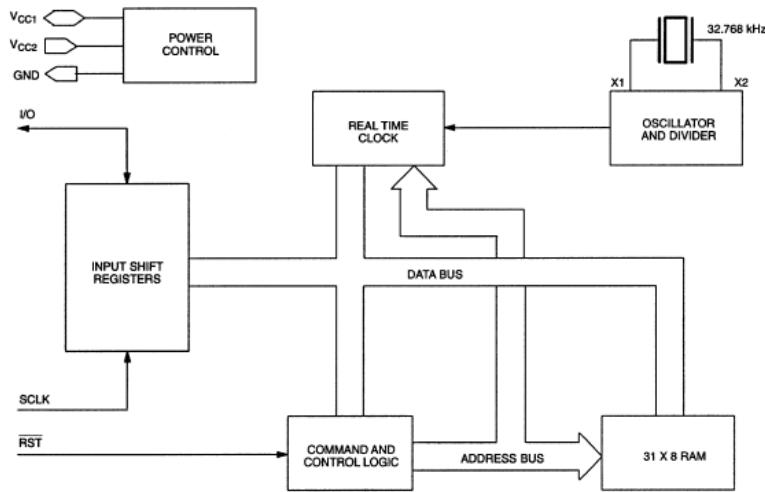
1. Tegangan kerja DC (12V).
2. Arus listrik (3A)
3. Auto Cut-off

4. Tekanan 100 Psi
5. Flow 4.0 LPM

## **K. RTC (Real Time Clock) DS1302**

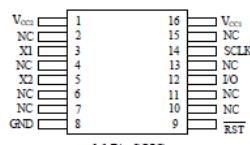
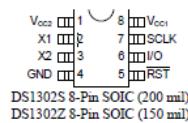
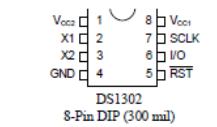
*Real Time Clock* merupakan suatu *chip* (IC) yang memiliki fungsi sebagai penyimpan waktu dan tanggal. DS1302 merupakan *Real Time Clock* yang dapat menyimpan data-data detik, menit, jam, hari, tanggal, bulan dan tahun dengan akurat. Adapun blok diagram dan pin map *Real Time Clock* DS1302 pada gambar 9 dan 10. Fitur dari *Real Time Clock* DS1302 antara lain:

1. Format waktu dapat disetting dalam format 12 jam (AM/PM) atau 24 jam
2. Memiliki kemampuan penyesuaian jumlah hari/bulan terhadap tahun kabisat
3. 31 x 8 Battery-Backed General-Purpose RAM
4. Menggunakan antarmuka I2C (Serial Data dan Serial Clock)
5. Menggunakan tegangan input kerja 5 Vdc
6. Harus memakai baterai back-up dengan range 2-5 Vdc
7. Konsumsi arus pada baterai back-up hanya 200nA
8. Range suhu kerja optimal antara -40°C sampai +85°C



**Gambar 9. Block Diagram *Real Time Clock DS1302***

Sumber: Datasheet DS1302



#### PIN DESCRIPTION

X1, X2	– 32.768 kHz Crystal Pins
GND	– Ground
RST	– Reset
I/O	– Data Input/Output
SCLK	– Serial Clock
Vcc1, Vcc2	– Power Supply Pins

**Gambar 10. Pin Map *Real Time Clock DS1302***

Sumber: Datasheet DS1302

## L. LCD (Liquid Crystal Display)

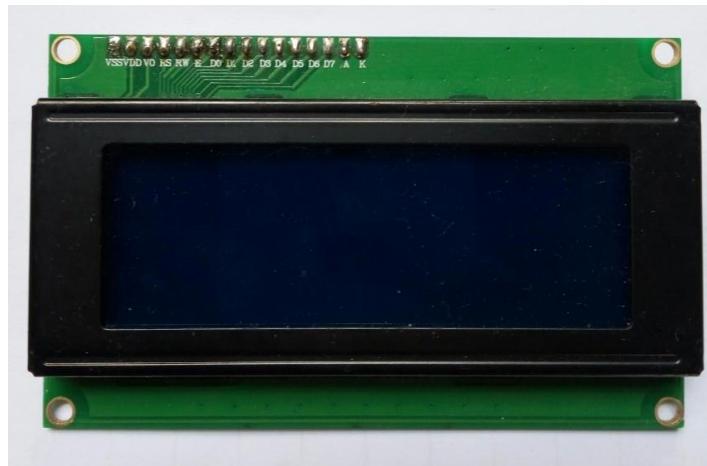
LCD (Liquid Crystal Display) adalah salah satu komponen elektronika yang berfungsi sebagai tampilan suatu data, baik karakter, huruf ataupun grafik. Dipasaran tampilan LCD sudah tersedia dalam bentuk modul yaitu tampilan LCD beserta rangkaian pendukungnya . LCD mempunyai pin data, kontrol catu daya, dan pengatur kontras tampilan.

LCD juga merupakan perangkat display yang paling umum dipasangkan di Mikrokontroller, mengingat ukurannya yang kecil dan kemampuan menampilkan karakter dan grafik yang lebih dibandingkan *seven segmen*. Didalam LCD terdapat beberapa register, seperti IR (Instruction Register), DR (Data Register), BF (Busy Flag), AC (Address Counter), DDRAM (Display Data Random Access Memory), CGROM (Character Generator Read Only Memory), CGRAM (Character Generator Random Access Memory), dan Cursor and Blink Control Circuit.

Adapun penggunaan LCD dalam proyek akhir ini adalah modul LCD 20 x 4 karakter. Salah satu alasan penggunaan modul LCD 20 x 4 adalah untuk menunjukan angka pengukuran. Untuk mengakses LCD 20 x 4 harus melakukan konfigurasi pin dari LCD dengan pin I/O mikrokontroller tersebut. Berikut Tabel 1 deskripsi pin pada LCD dan gambar 11 Modul LCD 20 x 4 Character.

**Tabel 1. Deskripsi pin pada LCD**

No	Symbol	Function
1	Vss	GND pin, 0 V
2	Vdd	Positive power pin, +5 V
3	Vo	LCD drive voltage input pin
4	Rs	Data/instruction select input pin
5	R/W	Read/Write select input pin
6	E	Enable input pin
7 - 14	D0 - D7	Data bus line
15	Led A	LED Power supply
16	Led K	LED Power supply



**Gambar 11. Modul LCD 20 x 4 Character**

## **M. FUZZY LOGIC CONTROLLER**

### **1. Pengantar Logika Fuzzy**

Munculnya logika fuzzy dilatarbelakangi oleh adanya sebuah kesenjangan antara hukum-hukum matematika dengan permasalahan sesungguhnya

dikehidupan nyata. Dengan demikian perlu suatu metode analisa baru untuk mendeteksi solusi yang optimal terhadap permasalahan real. Metode tersebut dikenal sebagai logika fuzzy (logika kabur atau tidak jelas). Ada beberapa alasan digunakannya logika fuzzy, antara lain:

- a. Konsep fuzzy logic mudah dimengerti dan fleksibel. Konsep matematis yang mendasari penalaran fuzzy sederhana dan mudah dimengerti.
- b. Logika fuzzy sangat fleksibel.
- c. Fuzzy logic memiliki toleransi terhadap data-data yang tidak tepat.
- d. Fuzzy logic mampu memodelkan fungsi-fungsi non-linear yang kompleks.
- e. Fuzzy logic dapat membangun dan mengaplikasikan pengalaman-pengalaman para pakar secara langsung tanpa harus melalui proses pelatihan.
- f. Fuzzy logic dapat bekerjasama dengan teknik-teknik kendali secara konvensional.
- g. Fuzzy logic didasarkan pada bahasa alami.

## 2. **Himpunan Fuzzy**

Pada himpunan tegas (Crisp), nilai keanggotaan suatu item  $x$  dalam suatu himpunan  $A$ , yang sering ditulis dengan  $A[x]$ , memiliki 2 kemungkinan, yaitu:

- a. satu (1), yang berarti bahwa suatu item menjadi anggota dalam suatu himpunan.
- b. nol (0), yang berarti bahwa suatu item tidak menjadi anggota dalam suatu himpunan.

Contoh:

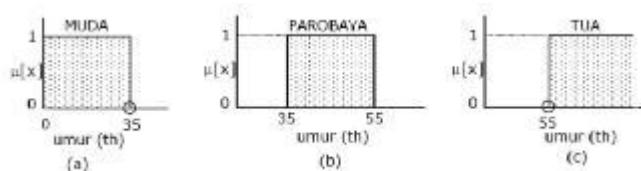
Misalkan variable umur dibagi menjadi 3 kategori, yaitu:

MUDA                       $umur < 35$  tahun

PAROBAYA                 $35 \geq umur \leq 55$  tahun

TUA                         $umur > 55$  tahun

Nilai keanggotaan secara grafis, himpunan MUDA, PAROBAYA, dan TUA ini dapat dilihat pada Gambar 12



**Gambar 12. Himpunan : MUDA, PAROBAYA, dan TUA**

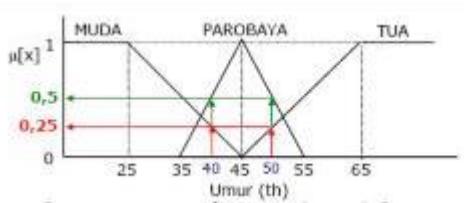
Sumber : Kusumadewi (2010,4)

Menurut Gambar dapat dilihat bahwa:

- a. apabila seseorang berusia 34 tahun, maka dapat dikatakan MUDA ( $MUDA[34]=1$ );
- b. apabila seseorang berusia 35 tahun, maka dapat dikatakan TIDAK MUDA ( $MUDA[35]=0$ );
- c. apabila seseorang berusia 35 tahun kurang 1 hari, maka dapat dikatakan TIDAK MUDA ( $MUDA[35 \text{ th} - 1 \text{ hr}]=0$ );

- d. apabila seseorang berusia 35 tahun, maka dapat dikatakan PAROBAYA ( $\text{PAROBAYA}[35]=1$ );
- e. apabila seseorang berusia 34 tahun, maka dapat dikatakan TIDAK PAROBAYA ( $\text{PAROBAYA}[34]=0$ );
- f. apabila seseorang berusia 36 tahun, maka dapat dikatakan PAROBAYA ( $\text{PAROBAYA}[36]=1$ );
- g. apabila seseorang berusia 35 tahun kurang 1 haru , maka dapat dikatakan TIDAK PAROBAYA ( $\text{PAROBAYA}[35 \text{ th} - 1 \text{ hr}]=0$ )

Dari uraian di atas dapat dikatakan bahwa pemakaian himpunan crips untuk menyatakan umur sangat tidak adil, adanya perubahan kecil saja pada suatu nilai mengakibatkan perbedaan kategori yang cukup signifikan. Himpunan fuzzy digunakan untuk mengantisipasi hal tersebut. Seseorang dapat masuk dalam 2 himpunan berbeda, MUDA dan PAROBAYA, PAROBAYA dan TUA, dsb. Seberapa besar eksistensinya dalam himpunan tersebut dapat dilihat pada gambar 13 nilai keanggotaannya.



**Gambar 13. Himpunan Fuzzy Untuk Variable Umur**

Sumber : Kusumadewi (2010,5)

- a. seseorang yang berumur 40 tahun, termasuk dalam himpunan MUDA dengan  $\mu_{MUDA}[40]=0.25$ ; Namun dia juga termasuk dalam himpunan PAROBAYA dengan  $\mu_{PAROBAYA}[40]=0.5$ .
- b. seseorang yang berumur 50 tahun, termasuk dalam himpunan TUA dengan  $\mu_{TUA}[50]=0.25$ ; Namun dia juga termasuk dalam himpunan PAROBAYA dengan  $\mu_{PAROBAYA}[50]=0.5$ .

Jika pada himpunan crips, nilai keanggotaan hanya ada 2 kemungkinan, yaitu 0 atau 1, maka pada himpunan fuzzy nilai keanggotaan terletak pada rentang 0 sampai 1. Apabila  $x$  memiliki nilai keanggotaan fuzzy  $A[x] = 0$  berarti  $x$  tidak menjadi anggota himpunan  $A$ , demikian pula apabila  $x$  memiliki nilai keanggotaan fuzzy  $A[x] = 1$  berarti  $x$  menjadi anggota penuh pada himpunan  $A$ . Himpunan fuzzy memiliki 2 atribut.

- a. Linguistik, yaitu penamaan suatu grup yang mewakili suatu keadaan atau kondisi tertentu dengan menggunakan bahasa alami, seperti : MUDA, PAROBAYA, TUA.
- b. Numeris, yaitu suatu nilai (angka) yang menunjukkan ukuran dari suatu variabel seperti: 40, 25, 50, dsb.

Ada beberapa hal perlu diketahui dalam memahami sistem fuzzy, yaitu :

- a. Variabel fuzzy

Variabel fuzzy merupakan variabel yang hendak dibahas dalam suatu sistem fuzzy. Contoh : Temperatur, Umur, Permintaan, dsb.

b. Himpunan fuzzy

Himpunan fuzzy merupakan suatu grup yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel fuzzy.

c. Semesta pembicaraan

Semesta pembicaraan merupakan kesuluran nilai yang diperbolehkan untuk dioperasikan dalam suatu variabel fuzzy. Semesta pembicaraan merupakan himpunan bilangan real yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan. Nilai semesta pembicaraan dapat berupa bilangan positif maupun negatif. Adakalanya nilai semesta pembicaraan ini tidak dibatasi batas atasnya.

d. Domain

Domain himpunan fuzzy merupakan kesuluran nilai yang diijinkan dalam semesta pembicaraan dan boleh dioperasikan dalam suatu himpunan fuzzy. Seperti halnya semesta pembicaraan, domain adalah himpunan bilangan real yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan. Nilai domain dapat berupa bilangan positif maupun negatif.

### 3. Fungsi Keanggotaan

Fungsi keanggotaan (Membership Function) adalah suatu kurva yang menunjukkan pemetaan titik-titik input data ke dalam nilai keanggotaannya (sering

juga disebut dengan derajat keanggotaan) yang memiliki interval antara 0 sampai

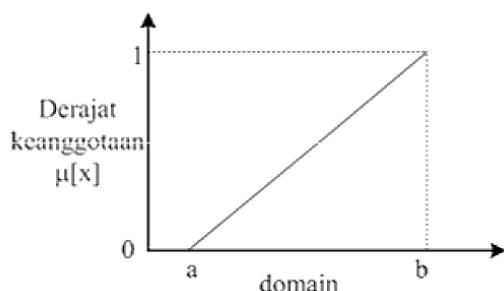
1. Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi.

Ada beberapa fungsi yang bisa digunakan diantaranya adalah sebagai berikut:

a. Representasi Linear

Pada representasi linear, pemetaan input ke derajat keanggotaannya digambarkan sebagai suatu garis lurus. Bentuk ini paling sederhana dan menjadi pilihan yang baik untuk mendekati suatu konsep yang kurang jelas.

Ada 2 keadaan himpunan fuzzy yang linear. Pertama, kenaikan himpunan dimulai pada nilai domain yang memiliki derajat keanggotaan nol [0] bergerak ke kanan menuju ke nilai domain yang memiliki derajat keanggotaan lebih tinggi seperti terlihat pada gambar 14.



**Gambar 14. Representasi Linear Naik**

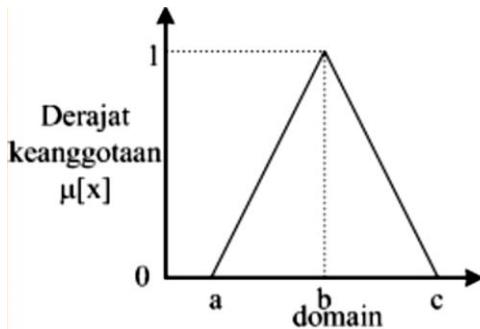
Sumber : Kusumadewi (2010,9)

Fungsi keanggotaan kurva linear naik :

$$\mu[x] = \begin{cases} 0 & , x < a \\ (x - a)/(b - a) & , a \leq x \leq b \\ 1 & , x > b \end{cases}$$

b. Representasi Kurva Segitiga

Kurva segitiga pada dasarnya merupakan gabungan antara 2 garis linear dan linear turun seperti terlihat pada gambar 15.



**Gambar 15. Kurva Segitiga**

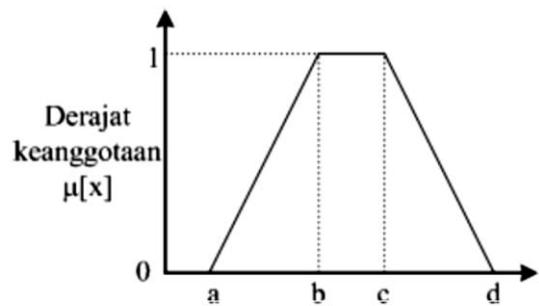
Sumber : Kusumadewi (2010,11)

Fungsi keanggotaan Kurva Segitiga

$$\mu[x,a,b,c] = \begin{cases} 0 & , x \leq a \text{ atau } x \geq c \\ (x - a)/(b - a) & , a \leq x \leq b \\ (c - x)/(c - b) & , b \leq x \leq c \end{cases}$$

### c. Representasi Kurva Trapesium

Kurva trapezium pada dasarnya seperti bentuk segitiga, hanya saja ada beberapa titik yang memiliki nilai keanggotaan 1. Gambar 16 menunjukkan kurva trapesium.



**Gambar 16. Kurva Trapesium**

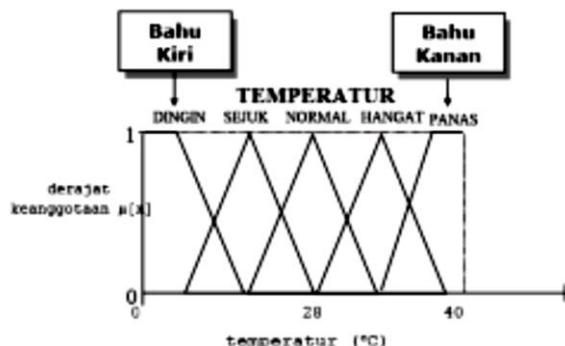
Sumber : Kusumadewi (2010,11)

Fungsi keanggotaan Kurva Trapesium

$$\mu[x,a,b,c,d] = \begin{cases} 0 & , x \leq a \\ \frac{(x-a)}{(b-a)} & , a \leq x \leq b \\ 1 & , b \leq x \leq c \\ \frac{(d-x)}{(d-c)} & , c \leq x \leq d \\ 0 & , x \geq d \end{cases}$$

#### d. Representasi Kurva Bentuk Bahu

Daerah yang terletak ditengah suatu variabel yang dipresentasikan dalam bentuk segitiga, pada sisi kanan dan kirinya akan naik dan turun (misalkan : DINGIN bergerak ke SEJUK bergerak ke HANGAT dan bergerak ke PANAS). Tetapi terkadang salah satu sisi dari variabel tersebut tidak mengalami perubahan. Sebagai contoh, apabila telah mencapai kondisi panas, kenaikan temperature akan tetap berada pada kondisi PANAS. Himpunan fuzzy ‘bahu’ bukan segitiga, digunakan untuk mengakhiri variabel suatu daerah fuzzy. Bahu kiri bergerak dari benar ke salah, demikian juga bahu kanan bergerak dari salah ke benar. Gambar 17 menunjukkan variabel TEMPERATUR dengan daerah bahunya.



**Gambar 17. Daerah Bahu pada Variabel TEMPERATUR**

Sumber : Kusumadewi (2010,14)

#### 4. Operator Dasar Zadeh untuk Operasi Himpunan Fuzzy

Seperti halnya himpunan konvensional, ada beberapa operasi yang didefinisikan secara khusus untuk mengkombinasi dan memodifikasi himpunan fuzzy. Nilai keanggotaan sebagai hasil dari operasi 2 himpunan sering dikenal dengan nama fire strength atau -predikat. Ada 3 operasi dasar yang diciptakan oleh Zadeh, yaitu :

a. Operator AND

Operator ini berhubungan dengan operasi interaksi pada himpunan. a-predikat sebagai hasil operasi dengan operator AND diperoleh dengan mengambil nilai keanggotaan terkecil antara elemen pada himpunan-himpunan yang bersangkutan.

$$\mu A \cap B = \min (\mu A[x], \mu B[y])$$

b. Operator OR

Operator ini berhubungan dengan operator union pada himpunan a-predikat sebagai hasil operasi dengan operator OR diperoleh dengan mengambil nilai keanggotaan terbesar antara elemen pada himpuna-himpunan yang bersangkutan.

$$\mu A \cup B = \max (\mu A[x], \mu B[y])$$

c. Operator NOT

Operator ini berhubungan dengan operasi komplemen pada himpunan a-predikat sebagai hasil operasi dengan operasi NOT diperoleh dengan

mengurangkan nilai keanggotaan elemen pada himpunan yang bersangkutan dari 1.

$$\mu A' = 1 - \mu A[x]$$

## 5. Sistem Inferensi Fuzzy

Sistem inferensi fuzzy adalah proses transformasi dari suatu input dalam domain fuzzy ke suatu output dalam domain fuzzy. Proses transformasi pada bagian inferensi membutuhkan aturan-aturan fuzzy yang terdapat didalam basis-basis aturan. Blok inferensi menggunakan teknik penalaran untuk menyeleksi basis-basis aturan dari basis pengetahuan. Metode inferensi yang sering digunakan dalam logika fuzzy adalah:

### a. Metode Tsukomoto

Pada Metode Tsukomot, setiap konsekuensi pada aturan yang berbentuk IF-THEN harus direpresentasikan dengan suatu himpunan fuzzy dengan fungsi keanggotaan yang monoton. Sebagai hasilnya, output hasil inferensi dari tiap-tiap aturan diberikan secara tegas (crisp) berdasarkan  $\alpha$ -predikat. Hasilnya akhirnya diperoleh dengan menggunakan rata-rata terbobot.

### b. Metode Sugeno

Penalaran dengan metode Sugeno hampir sama dengan penalaran Mamdani, hanya saja output (konsekuensi) sistem tidak berupa himpunan fuzzy,

melainkan berupa konstanta atau persamaan linear. Metode ini diperkenalkan oleh Takagi-Sugeno Kang pada tahun 1985, sehingga metode ini sering juga dinamakan dengan metode TSK. Metode TSK terdiri dari 2 jenis, yaitu:

- 1) Model Fuzzy Sugeno Orde-Nol

Secara umum bentuk model fuzzy Sugeno Orde-Nol adalah:

$$\text{IF}(x_1 \text{ is } A_1) (x_2 \text{ is } A_2) (x_3 \text{ is } A_3) \dots (x_N \text{ is } A_N) \text{ THEN } z=k$$

Dengan  $A_1$  adalah himpunan fuzzy ke-1 sebagai antecedent, dan  $k$  adalah suatu konstanta (tegas) sebagai konsekuen.

- 2) Model Fuzzy Sugeno Orde-Satu

Secara umum bentuk model fuzzy Sugeno Orde-Satu adalah:

$$\text{IF } (x_1 \text{ is } A_1) \dots (x_N \text{ is } A_N) \text{ THEN } z=p_1*x_1+\dots+p_N*x_N+q$$

Dengan  $A_1$  adalah himpunan fuzzy ke-1 sebagai antecedent, dan  $p_1$  adalah konstanta (tegas) ke-1 dan  $q$  juga merupakan konstanta dalam konsekuen. Apabila komposisi aturan menggunakan Sugeno, maka defuzzifikasi dilakukan dengan cara mencari nilai rata-ratanya.

### c. Metode Mamdani

Metode Mamdani sering juga dikenalkan dengan nama Metode Max Min. Metode ini diperkenalkan oleh Ebrahim Mamdani pada tahun 1975. Untuk mendapatkan output, diperlukan 4 tahapan.

1) Pembentukan himpunan fuzzy

Pada metode Mamdani, baik variabel *input* maupun variabel *output* dibagi menjadi satu atau lebih himpunan fuzzy.

2) Aplikasi fungsi implikasi

Pada metode Mamdani, fungsi implikasi yang digunakan adalah Min.

3) Komposisi aturan

Tidak seperti penalaran monoton, apabila sistem terdiri dari beberapa aturan, maka inferensi diperoleh dari kumpulan dan korelasi antar aturan. Ada 3 metode yang digunakan dalam melakukan inferensi sistem fuzzy, yaitu: Max, Additive, dan Propabilistic OR (probior).

a) Metode Max (Maximum)

Pada metode ini, solusi himpunan fuzzy diperoleh dengan cara mengambil nilai maksimum aturan, kemudian menggunakan untuk memodifikasi daerah fuzzy dan mengaplikasikannya ke *output* dengan menggunakan operator OR (union). Jika semua proposisi telah dievaluasi, maka *output* akan berisi suatu himpunan fuzzy yang merefleksikan kontribusi dari tiap-tiap proposisi. Secara umum dapat dituliskan:

$$\mu_{sf[xi]} = \max (\mu_{sf[xi]}, \mu_{kf[xi]})$$

Dengan:

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

$\mu_{kf[xi]}$  = nilai keanggotaan konsekuensi fuzzy hingga aturan ke-*i*;

b) Metode Additive (Sum)

Pada metode ini, solusi himpunan fuzzy diperoleh dengan cara melakukan bounded-sum terhadap semua output daerah fuzzy. Secara umum dituliskan:

$$\mu_{sf[xi]} = \min (1, \mu_{sf[xi]} + \mu_{kf[xi]})$$

Dengan:

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

$\mu_{kf[xi]}$  = nilai keanggotaan konsekuensi fuzzy hingga aturan ke-*i*;

c) Metode Probabilistik OR (probor)

Pada metode ini, solusi himpunan fuzzy diperoleh dengan cara melakukan product terhadap semua output daerah fuzzy. Secara umum dituliskan:

$$\mu_{sf[xi]} = (\mu_{sf[xi]} + \mu_{kf[xi]}) - (\mu_{sf[xi]} * \mu_{kf[xi]})$$

Dengan:

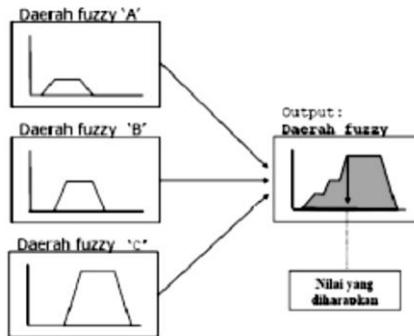
$\mu_{sf[xi]}$  = nilai keanggotaan solusi fuzzy sampai aturan ke-*i*;

$\mu_{kf[xi]}$  = nilai keanggotaan konsekuensi fuzzy hingga aturan ke-*i*;

4) Penegasan (Defuzzy)

*Input* dari proses defuzzifikasi adalah suatu himpunan fuzzy yang diperoleh dari komposisi aturan-aturan fuzzy, sedangkan *output* yang dihasilkan merupakan suatu bilangan pada domain himpunan fuzzy tersebut. Sehingga

jika diberikan suatu himpunan fuzzy dalam range tertentu, maka harus dapat diambil suatu nilai crisp tertentu sebagai output seperti terlihat pada gambar 18.



**Gambar 18. Proses Defuzzifikasi**

Sumber : Kusumadewi (2010,40)

Ada beberapa metode defuzzifikasi pada komposisi aturan Mamdani, antara lain :

- a) Metode Centroid (Composite Moment)

Pada metode ini, solusi crisp diperoleh dengan cara mengambil titik pusat ( $z^*$ ) daerah fuzzy. Secara umum dirumuskan:

$$z^* = \frac{\int_z z \cdot \mu(z) dz}{\int_z \mu(z) dz}, \text{ untuk semesta kontinu}$$

$$z^* = \frac{\sum_{j=1}^n z_j \mu(z_j)}{\sum_{j=1}^n \mu(z_j)}, \text{ untuk semesta diskret}$$

b) Metode Bisektor

Pada metode ini, solusi crisp diperoleh dengan cara mengambil nilai pada domain fuzzy yang memiliki nilai keanggotaan setengah dari jumlah nilai keanggotaan daerah fuzzy. Secara umum dituliskan:

$$Z_p \text{ sedemikian hingga } \int_1^p \mu(z) dz = \int_p^n \mu(z) dz$$

c) Metode Mean of Maksimum (MOM)

Pada metode ini, solusi crisp diperoleh dengan cara mengambil nilai rata-rata domain yang memiliki nilai keanggotaan maksimum.

d) Metode Largest of Maksimum (LOM)

Pada metode ini, solusi crisp diperoleh dengan cara mengambil nilai terbesar dari domain yang memiliki nilai keanggotaan maksimum.

e) Metode Smallest of Maksimum (SOM)

Solusi crisp diperoleh dengan cara mengambil nilai terkecil dari domain yang memiliki nilai keanggotaan maksimum.

## 6. Fuzzy Reasoning

Fuzzy reasoning adalah suatu prosedur inferensi yang menghasilkan kesimpulan dari satu set if-then-rules dan fakta yang diketahui. Misalkan:

A = Hari ini cerah

$A \rightarrow B$  : Hari = cerah maka langit = biru

Inferensi

: Langit biru

Premis 1 (Fakta)

: Hari ini cerah ( $x=A$ )

Premis 2 (Rule)

: Jika hari ini cerah maka langit berwarna biru

(if  $x = A$  then  $y=B$ )

Kesimpulan

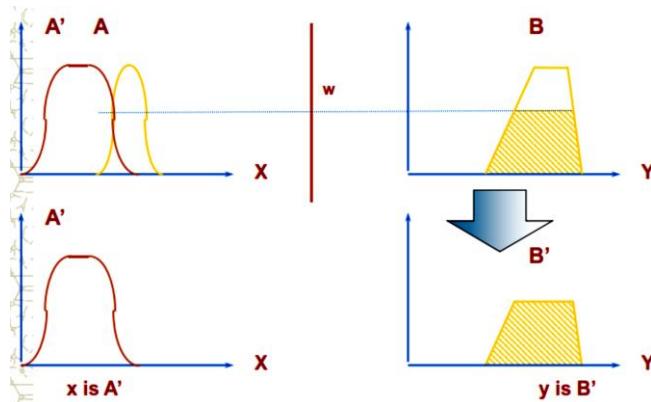
: Hari ini langit berwarna biru ( $x = B$ )

Diasumsikan bahwa implikasi fuzzy  $A \rightarrow B$  dinyatakan sebagai  $R = x * y$ .

Maka himpunan fuzzy  $B$  disebabkan oleh  $x$  dan rule “if  $x = A$  then  $y = B$ ”

didefinisikan sebagai  $\mu_B(y) = \max\min[\mu_A'(x), \mu_R(x,y)]$ .

a. Single Rule Single Antecedent



**Gambar 19. Fuzzy Reasoning Single Rule Single Antecedent**

Sumber : J.S.R Jang, C.T Sun, E.Mizutani (1997)

Gambar 19 merupakan gambaran proses Fuzzy reasoning single rule single antecedent.

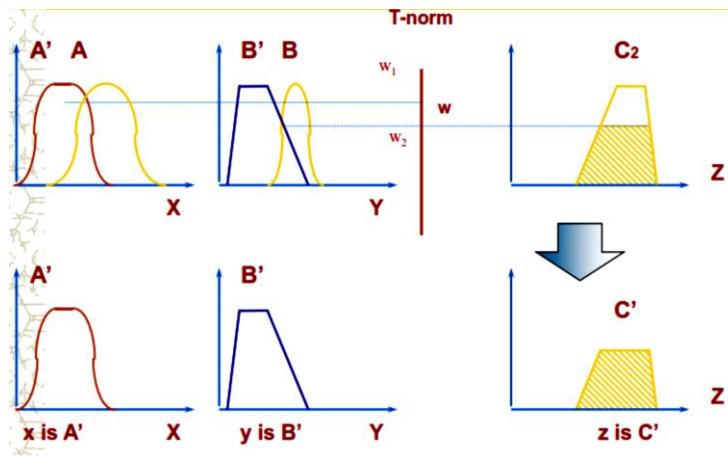
Rule : if  $x$  adalah  $A$  maka  $Y$  adalah  $B$

Fakta :  $x$  adalah  $A$

Kesimpulan :  $y \text{ is } B' (\mu B'(y) = [\mu x(\mu A'(x) \cap \mu A(x))] \cap \mu B(y))$

Untuk lebih jelasnya dapat dilihat pada gambar diatas.

b. Single Rule Multiple Antecedent



**Gambar 20. Fuzzy Reasoning Single Rule Multiple Antecedent**

Sumber : J.S.R Jang, C.T Sun, E.Mizutani (1997)

Gambar 20 merupakan gambaran proses Fuzzy reasoning single rule multiple antecedent.

Fakta :  $x \text{ adalah } A' \text{ dan } y \text{ adalah } B'$

Rule : Jika  $x$  adalah  $A$  dan  $y$  adalah  $B$  maka  $z$  adalah  $C$

Kesimpulan :  $z$  adalah  $C'$

Rule :  $A * B \rightarrow C$

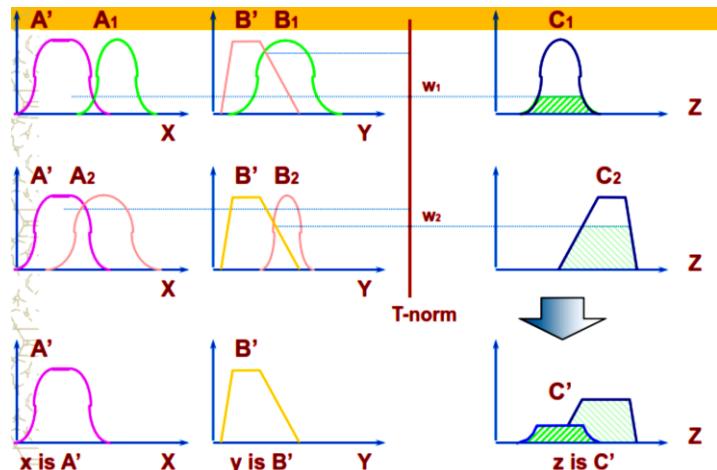
$$R_{mamdan} (A, B, C) = (A * B) * C = \int_{X * Y * Z} \mu A(x) \wedge \mu B(y) \wedge \mu C(z) / (x, y, z)$$

$C' = \text{Premise 1} \circ \text{Premise 2}$

$$= (A' * B') \circ (A * B \rightarrow C)$$

$$\begin{aligned}\mu_C(z) &= \bigvee_{x,y} [\mu_{A'}(x) \wedge \mu_{B'}(y)] \wedge [\mu_A(x) \wedge \mu_B(y) \wedge \mu_C(z)] \\ &= \bigvee_{x,y} \{ [\mu_{A'}(x) \wedge \mu_{B'}(y)] \wedge [\mu_A(x) \wedge \mu_B(y)] \} \wedge \mu_C(z) \\ &= \{ \bigvee_x [\mu_{A'}(x) \wedge \mu_A(x)] \} \wedge \{ \bigvee_y [\mu_{B'}(y) \wedge \mu_B(y)] \} \wedge \mu_C(z)\end{aligned}$$

c. Multiple Rules Multiple Antecedent



**Gambar 21. Fuzzy Reasoning Multiple Rules Multiple Antecedent**

Sumber : J.S.R Jang, C.T Sun, E. Mizutani (1997)

Gambar 21 merupakan gambaran proses Fuzzy reasoning multiple antecedent.

Fakta : x adalah A' dan y adalah B'

Rule 1 : Jika x adalah A1 dan y adalah B1 maka z adalah C1

Rule 2 : Jika x adalah A2 dan y adalah B2 maka z adalah C2

Kesimpulan : z adalah C'

Rule 1 =  $A_1$  dan  $B_1 \rightarrow C_1$

Rule 2 =  $A_2$  dan  $B_2 \rightarrow C_2$

Karena max-min operator komposisi  $\circ$  adalah distributive atas operator union, maka :

$$\begin{aligned} C' &= (A'^*B') \circ (R_1 \cup R_2) = [(A'^*B') \circ R_1] \cup [(A'^*B') \circ R_2] \\ &= C'_1 \cup C'_2 \end{aligned}$$

Dapat di lihat dimana  $C'_1$  dan  $C'_2$  adalah himpunan fuzzy tereke untuk aturan 1 dan 2 masing-masing.

## **BAB III**

### **KONSEP RANCANGAN**

Pembuatan dan pengimplementasian menggunakan metode **ADDIE** (Dick dan Carey,2010) pada proses rancang bangunnya, sedangkan dalam pengaplikasiannya metode ADDIE (*Analysis, Design, Development, Implementation, and Evaluation*) merupakan metode yang berisi jajaran mengenai proses generic dalam melakukan suatu desain instruksional. Metode ini menerapkan pedoman deskriptif untuk mendukung kinerja yang efektif dalam 5 langkah yaitu *Analysis* yang bertujuan untuk melakukan identifikasi kebutuhan yang diperlukan. *Design* yang bertujuan untuk merancang dan merekonstruksi gambaran perencanaan dari alat baik berupa desain mekanik, desain elektronik dan desain perangkat lunak. *Development* bertujuan untuk pengembangan lebih lanjut atau perealisasian dari desain mekanik, desain elektronik dan desain perangkat lunak. *Implementation* bertujuan untuk mensinergikan dari tiap mekanik, elektronik dan perangkat lunak. Sedangkan yang terakhir *Evaluation* yang bertujuan untuk mengevaluasi segala kekurangan dan kelebihan dari proyek akhir ini untuk didapatkan kinerja yang lebih optimal.

#### **A. ANALYSIS**

Analisis pada proyek akhir ini terbagi menjadi rincian mengenai kebutuhan alat dan bahan guna menunjang pembuatan alat, pada tahap analisis ini dibagi

menjadi 2 yaitu identifikasi kebutuhan dan analisis kebutuhan, ditiap identifikasi kebutuhan dan analisa kebutuhan akan dibahas secara lengkap mengenai *hardware* dan *software* yang digunakan pada proyek akhir ini.

### **1. Identifikasi Kebutuhan**

Pada pengerjaan proyek akhir ini, terdapat beberapa kebutuhan untuk dapat menunjang pengerjaan proyek akhir ini yang dibagi menjadi *Hardware* dan *Software*:

#### **a. *Hardware***

Pada bagian *Hardware* akan dibagi menjadi 2 yaitu perancangan mekanik dan elektronik alat proyek akhir yang dijelaskan sebagai berikut:

##### **1) Mekanik**

1. Frame alat
  - Besi siku
  - Mur dan baut
2. Penyiraman Air dan Pestisida
  - Pipa paralon
  - Tabung Penampung
  - Selang Air
3. Panel Box
  - Plat Besi

## 2) **Elektronik**

1. Shield Mikrokontroller
2. *Power Supply*
3. Modul *Step Down*
4. LCD 20x4
5. Sensor Suhu (DS18B20)
6. Sensor Kelembaban Tanah
7. Relay
8. Modul Real Time Clock
9. Selenoid Valve
10. Pompa Air DC

### **b. *Software***

1. Aplikasi Pemrograman Logika Fuzzy (Matlab)
2. Aplikasi Pemrograman Mikrokontroller

## **2. Analisis Kebutuhan**

Analisis kebutuhan mengacu pada identifikasi kebutuhan yang telah dipaparkan sebelumnya, berikut merupakan beberapa analisis kebutuhan pada bagian *hardware* maupun *software* guna menunjang penggerjaan proyek akhir:

### **a. *Hardware***

#### **1) *Mekanik***

### 1. Frame Alat

Dalam perancangan alat proyek akhir ini, frame alat digunakan sebagai media peletakan seluruh komponen proyek akhir.

- Besi siku yang berfungsi sebagai bahan dasar kerangka frame alat pada proyek akhir ini.
- Mur dan baut yang berfungsi sebagai perangkat untuk mengencangkan antara besi siku.

### 2. Penyiraman Air dan Pestisida

- Pipa Paralon ukuran  $\frac{1}{2}$  inci yang berfungsi sebagai jalur penyiraman air yang terhubung dengan katup.
- Tabung Penampung ukuran 5 inci yang berfungsi sebagai tempat penampungan air dan pestisida.
- Selang air yang berfungsi sebagai jalur air dan pestisida dari penampung ke penyiraman.

### 3. Panel Box

- Plat besi berfungsi sebagai bahan dasar dari panel box.

## 2) Elektronik

Perancangan Elektronik dibutuhkan untuk menunjang alat, secara umum perancangan elektronik disesuaikan dengan identifikasi kebutuhan yang diperlukan pada tiap komponen seperti yang telah dipaparkan di atas:

## 1. Shield Mikrokontroller

Di dalam sebuah mikrokontroller terdapat sebuah IC yang berguna untuk mengontrol dan mengolah data namun jalur pada IC kecil. Maka dari itu shield ini digunakan untuk mempermudah perkabelan pada mikrokontroller ke komponen elektronik lain.

## 2. Power Supply

Rangkaian *power supply* dibutuhkan untuk memberikan catu daya ke rangkaian elektronik pada alat. Pada proyek akhir ini membutuhkan *supply* tegangan 12 volt DC.

## 3. Modul Step Down

Modul *Step Down* dibutuhkan untuk menurunkan tegangan dari 12 volt ke 5 volt sebagai catu daya mikrokontroller.

## 4. LCD 20x4

Dibutuhkan sebuah Liquid Crystal Display (LCD) untuk memonitoring program yang sedang berjalan secara langsung, menggunakan LCD agar program yang dijalankan termonitoring dengan baik.

## 5. Sensor Suhu

Sensor suhu pada proyek akhir ini menggunakan sensor DS18B20 yaitu sensor suhu yang menggunakan digital sebagai *output* dan mempunyai akurasi yang tinggi.

## 6. Sensor Kelembaban Tanah

Sensor kelembaban tanah digunakan untuk mendeteksi kadar kelembaban tanah di sekitar pot tanaman. Sensor kelembaban tanah digunakan karena sensor tersebut mempunyai kecepatan yang cepat dalam mendeteksi kadar kelembaban tanah.

## 7. Relay

*Relay* digunakan sebagai saklar untuk mengaktifkan dan menonaktifkan pompa air dan *solenoid valve*. Tegangan *input* relay adalah 5 volt DC.

## 8. Modul Real Time Clock

Modul *Real Time Clock* menggunakan IC type DS1302 yang digunakan sebagai penyimpan dan pengatur waktu (detik, menit, jam, hari, bulan dan tahun).

## 9. Selenoid Valve

*Solenoid valve* digunakan sebagai kran / katup untuk penyiraman air yang mempunyai tegangan kerja 220 volt.

## 10. Pompa Air DC

Pompa Air DC menggunakan tegangan kerja sebesar 12 volt serta mempunyai daya 4,8 watt digunakan sebagai pendorong air dan pestisida dari penampung ke tempat penyiraman.

### b. *Software*

Pada bagian *software* dibagi menjadi 2 bagian yaitu pemrograman Logika Fuzzy dan pemrograman Mikrokontroller

#### 1. Aplikasi Pemrograman Logika Fuzzy (Matlab)

Dalam pemrograman logika fuzzy pada proyek akhir ini menggunakan bantuan Matlab terlebih dahulu sebagai penentuan fuzzyifikasi, rule fuzzy, dan defuzzyifikasi. Pada rule fuzzy digunakan metode mamdani sedangkan untuk defuzzyifikasi digunakan metode *center of area*.

#### 2. Aplikasi Pemrograman Mikrokontroller

Pemrograman Mikrokontroller menggunakan *software* Arduino IDE sebagai *compiler* bahasa C yang digunakan untuk IC ATMega. Arduino IDE digunakan sebagai pengolahan sensor dan program logika fuzzy. Arduino IDE

dipilih karena proses program yang sangat detail pada *library* sehingga dapat mempermudah pembacaab sensor serta merupakan program *open source*.

## **B. DESIGN**

### **1. *Hardware***

#### **a. Perancangan Mekanik**

Pada bagian perancangan mekanik dibagi menjadi 3 bagian yaitu perancangan frame alat, penyiraman air dan pestisida, serta panel box.

##### **1) Perancangan Frame Alat**

Frame alat menggunakan besi siku dengan panjang 63 cm, lebar 36 cm, dan tinggi 80 cm. pemilihan bahan besi siku karena bersifat kokoh dan kuat untuk menopang seluruh komponen proyek akhir serta harganya lebih terjangkau. Pada frame alat terdapat bagian penyiraman, tempat tanaman, tempat penampungan, dan tempat panel box yang ditunjukan pada Gambar 22.



**Gambar 22. Desain Frame Alat**

2) Perancangan Penyiraman Air dan Pestisida



**Gambar 23. Desain Penyiraman Air dan Pestisida**

Gambar 23 merupakan desain penyiraman air dan pestisida yang telah digabungkan dengan frame. Pada penyiraman air digunakan pipa paralon ukuran  $\frac{1}{2}$  inci dan selang sebagai media penyaluran air sedangkan penyiraman pestisida digunakan selang. Penampung air dan pestisida menggunakan pipa ukuran 5 inci dengan kapasitas masing-masing 3 liter.

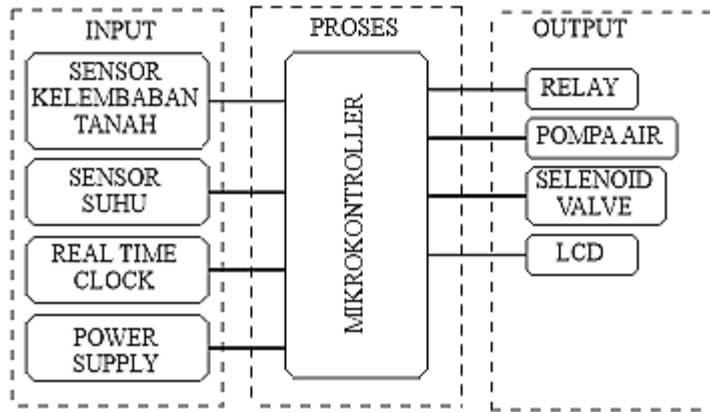
### 3) Perancangan Panel Box



**Gambar 24. Desain Panel Box**

Gambar 24 merupakan desain panel box dengan ukuran tinggi 33 cm, lebar 12 cm, dan panjang 20 cm. Panel box didesain sedemikian rupa agar dapat melindungi komponen elektronik penting proyek akhir ini. Pemilihan plat alumunium ketebalan 3 mm bertujuan agar tidak mudah rusak ketika terkena benturan .

## b. Perancangan Elektronik



**Gambar 25. Blok Diagram Sistem**

Gambar 25 merupakan blok diagram sistem yang terdiri dari *input*, *proses*, dan *output*. Berikut ini adalah perancangan elektronik yang terdapat pada blok diagram.

### 1) Shield Mikrokontroller

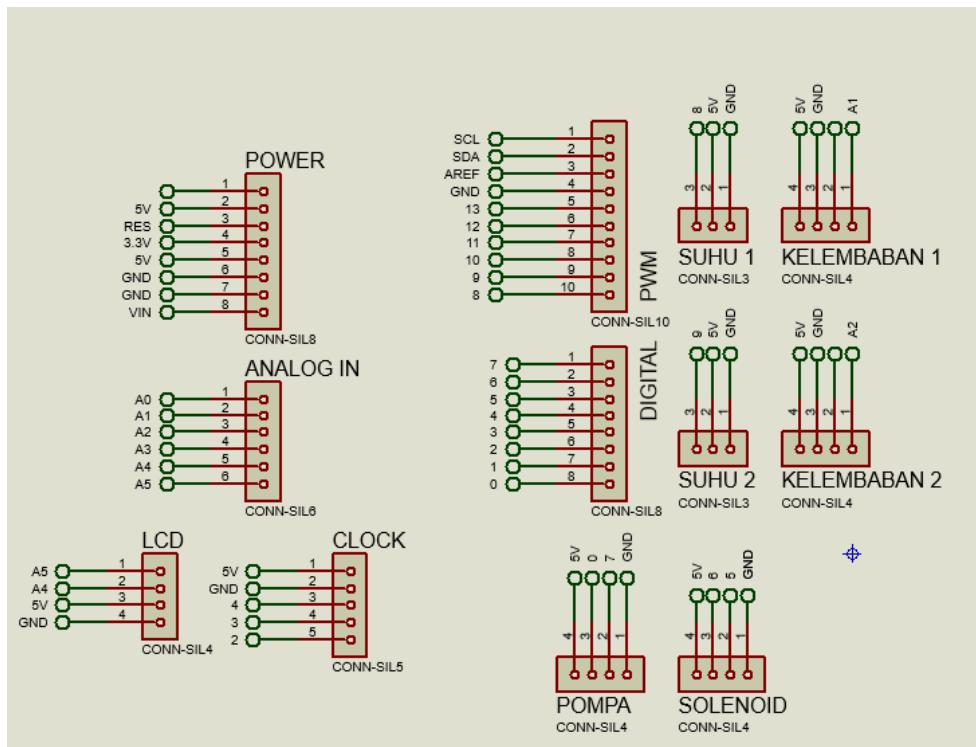
Mikrokontroller yang dibuat membutuhkan desain shield yang bertujuan agar dapat bekerja sesuai dengan kebutuhan dari alat tersebut dan untuk merapikan perkabelan pada alat agar tidak mengganggu mekanik pada alat tersebut. Desain shield mikrokontroller menggunakan soket *white housing* pin 4, pin 6, pin 8 dan pin 10. Berikut ini Tabel 2 kebutuhan *input* dan *output* Atmega dari alat ini.

**Tabel 2. Kebutuhan I/O ATMega**

Kebutuhan	Keterangan
Sensor Suhu	2 PIN Digital (2 Sensor)
Sensor Kelembaban Tanah	2 PIN Analog (2 Sensor)
Modul Real Time Clock	3 PIN Digital

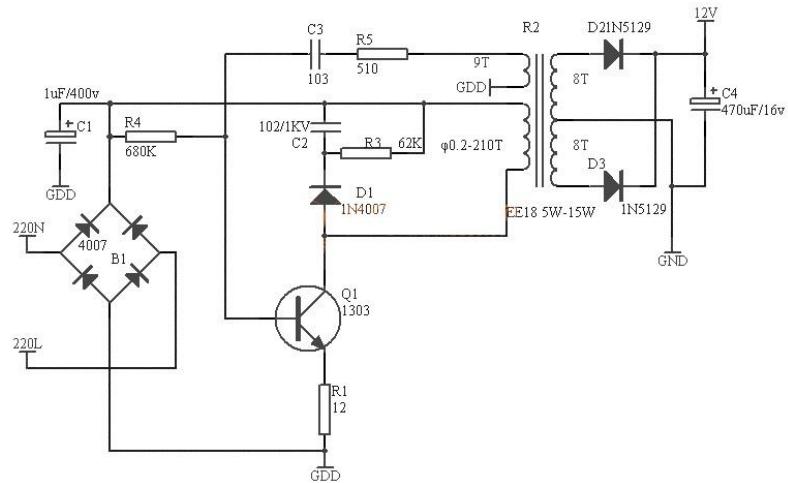
Relay	4 PIN Digital (3 Motor)
LCD 4x20	7 PORT
Total	18 I/O

Dari Tabel 2 kebutuhan I/O ATMega agar mempermudah perkabelan pada alat maka di desain sebuah shield untuk mempermudah perkabelan pada alat seperti pada Gambar 26.



**Gambar 26. Shield Arduino Uno**

## 2) Power Supply

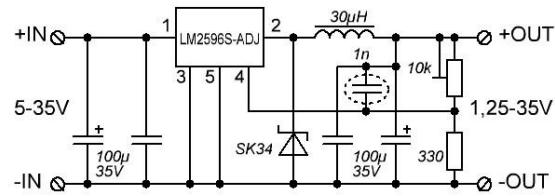


**Gambar 27. Power Supply**

Gambar 27 adalah rangkaian *power supply* dengan *input* 220 volt AC dan menghasilkan *output* 12 volt DC . Terdapat *fuse* pada rangkaian *power supply* di atas yang berfungsi sebagai pengaman tegangan apabila terjadi kesalahan tegangan.

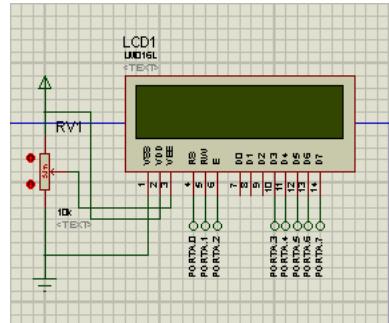
## 3) Modul Step Down

Rangkaian modul *step down* menggunakan IC LM2596 dengan *input* 12 volt DC dan menghasilkan *output* sesuai kebutuhan. Pada proyek akhir ini modul *step down* mengeluarkan *output* 5 volt DC sebagai *input* tegangan mikrokontroller dan sensor. Berikut ini Gambar 28 rangkaian modul *step down*.



**Gambar 28. Rangkaian modul *step down***

4) LCD 20x4

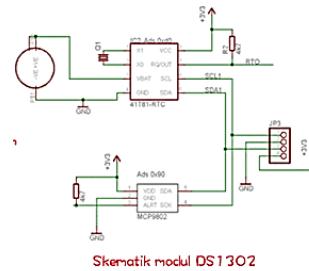


### **Gambar 29. Rangkaian LCD 20x4**

Gambar 29 merupakan rangkaian LCD 20x4 digunakan sebagai tampilan informasi dari program yang dibuat, ketika program berjalan maka program dikeluarkan ke dalam bentuk tulisan pada LCD agar dapat mengetahui informasi pada alat.

## 5) Modul Real Time Clock

Modul *Real Time Clock* menggunakan IC tipe DS1302 sebagai pengatur dan penyimpan waktu . Modul *Real Time Clock* digunakan sebagai *input* waktu mikrokontroller. Berikut ini Gambar 30 Modul *Real Time Clok*.



**Gambar 30. Modul Real Time Clock**

## 2. Software

### a. Pemrograman Logika Fuzzy (Matlab)

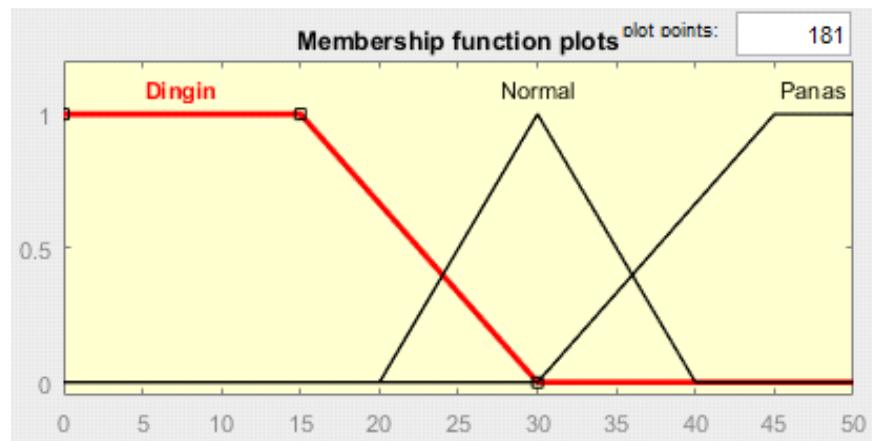
#### 1) Fuzzyfikasi

Fuzzyfikasi adalah suatu metode yang digunakan untuk mengolah suatu masukan dari bentuk tegas (*crisp*) menjadi fuzzy (variable linguistic), pada dasarnya disajikan berupa bentuk himpunan fuzzy dengan suatu fungsi keanggotaanya masing-masing dari masukan (*input*). Pada proyek akhir ini terdapat 2 variable yaitu variabel suhu dan variabel kelembaban tanah. Variabel suhu mempunyai 3 buah linguistic *value* yaitu dingin, normal dan panas. Sedangkan variabel kelembaban tanah mempunyai 3 buah linguistik value yaitu basah, normal dan kering. Pada bagian output terdapat variabel 3 buah linguistic *value* yaitu sebentar, normal dan lama.

##### – Variabel Suhu

Variabel Suhu dibuat dengan range pengukuran dari 0°C sampai dengan 50°C. Nilai ini diperoleh dari mencari rata-rata dari suhu maksimal dan suhu

minimal dari tanaman . Adapun gambar di bawah ini merepresentasikan nilai untuk tiap fungsi keanggotaan input suhu. Pada sistem Fuzzy tersebut, dijabarkan menjadi 3 buah membership function yaitu dingin, normal, dan panas. Berikut ini Gambar 31 Fungsi Keanggotaan *input* suhu.



**Gambar 31. Fungsi Keanggotaan *input* suhu**

Contoh :

Pada saat suhu diperoleh dan dikemudian diolah, kita asumsikan bahwa suhu yang diperoleh adalah  $25^{\circ}\text{C}$ , maka lihat pada Gambar 32 bahwa  $24^{\circ}\text{C}$  masuk pada kondisi



**Gambar 32. Contoh Kondisi Fuzzyfikasi Suhu**

Dapat dilihat bahwa  $25^{\circ}\text{C}$  berada antara Dingin dan Normal, jika kita kalkulasikan secara manual maka akan diperoleh seperti berikut ini:

$$x = 25$$

$$u_{\text{dingin}} = (d - x) / (d - c)$$

$$= (30 - 25) / (30 - 15)$$

$$= 5 / 15$$

$$= 0.333$$

$$u_{\text{normal}} = (x - a) / (b - a)$$

$$= (25 - 20) / (30 - 20)$$

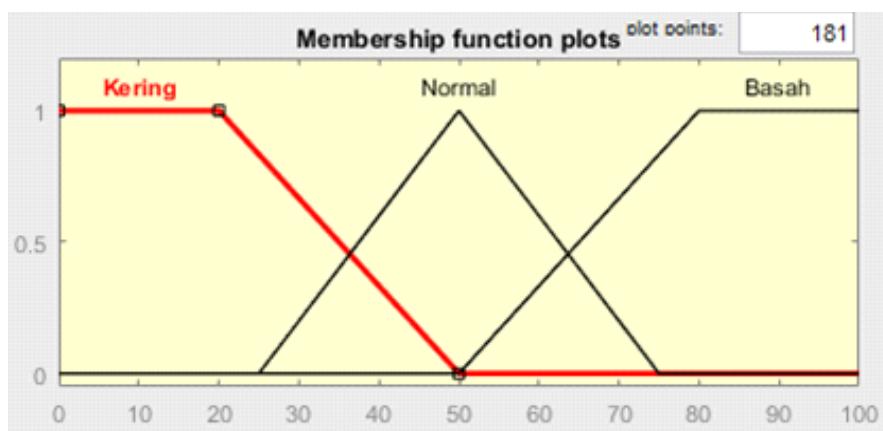
$$= 5 / 10$$

$$= 0.5$$

Berdasarkan perhitungan di atas dapat disimpulkan bahwa  $25^{\circ}\text{C}$  memiliki derajat keanggotaan sebesar 0.333 pada fungsi keanggotaan dingin dan 0.5 pada fungsi keanggotaan normal.

– Variabel Kelembaban Tanah

Variabel Kelembaban Tanah memiliki range mulai dari 0 sampai dengan 100. Nilai tersebut diperoleh dari mencari rata-rata dari hasil kelembaban tanah yaitu kering, normal, dan basah. Adapun fungsi keanggotaan dalam bentuk membership function dapat dilihat pada gambar 33:



**Gambar 33. Fungsi Keanggotaan Input Kelembaban Tanah**

Contoh :

Asumsikan bahwa kelembaban tanah yang terdeteksi oleh sensor adalah 40, maka bisa dilihat pada gambar 34 , bahwa 40 masuk pada kondisi:



**Gambar 34. Contoh kondisi Fuzzyifikasi Kelembaban Tanah**

Dapat dilihat bahwa 40 berada antara Kering dan Normal, jika kita kalkulasikan secara manual maka akan diperoleh seperti berikut ini:

$$y = 40$$

$$u_{\text{kering}} = (d - x) / (d - c)$$

$$= (50 - 40) / (50 - 20)$$

$$= 10 / 30$$

$$= 0.333$$

$$u_{\text{normal}} = (x - a) / (b - a)$$

$$= (40 - 25) / (50 - 25)$$

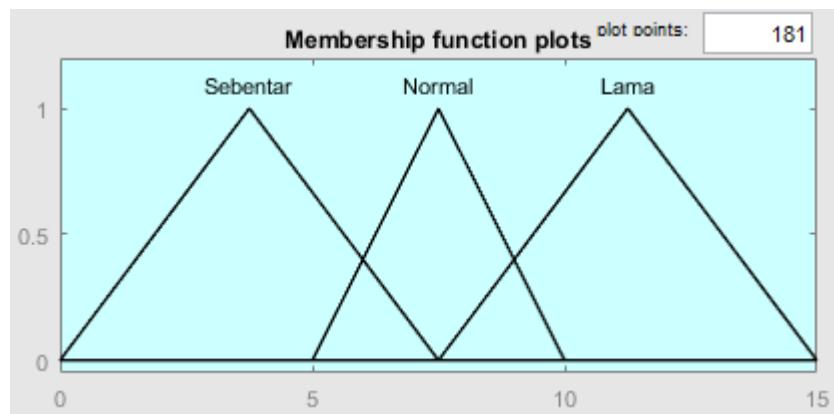
$$= 15 / 25$$

$$= 0.6$$

Berdasarkan perhitungan diatas dapat disimpulkan bahwa kelembaban tanah 40 memiliki derajat keanggotaan sebesar 0.333 pada fungsi keanggotaan kering dan 0.6 pada fungsi keanggotaan normal.

- Variabel Waktu (*timer*)

Variabel Waktu (*timer*) merupakan hasil akhir atau output dari perhitungan variabel suhu dan variabel kelembaban tanah, dibuat dengan range pengukuran dari 0 detik sampai dengan 15 detik. Pada variabel waktu (*timer*) memiliki 3 buah *linguistik value* diantaranya sebentar, normal, dan lama. Adapun Gambar 35 merepresentasikan nilai untuk tiap fungsi keanggotaan input waktu (*timer*).



**Gambar 35. Fungsi Keanggotaan Variabel Waktu**

## 2) Fuzzy Rule

Fuzzy Rule adalah suatu aturan yang dibuat untuk memenuhi syarat dari sebuah output yang terjadi. Aturan yang digunakan pada proyek akhir ini mengacu

pada penelitian yang sudah dilakukan oleh Bayu Aji Kurniawan pada tahun 2015, aturan tersebut dilustrasikan pada Tabel 3 berikut ini.

**Tabel 3. Fuzzy Rule**

Kelembaban / Suhu	Suhu Dingin	Suhu Normal	Suhu Panas
Kelembaban Kering	Lama (R1)	Lama (R2)	Lama(R3)
Kelembaban Normal	Sebentar (R4)	Normal (R5)	Normal (R6)
Kelembaban Basah	Sebentar (R7)	Sebentar (R8)	Sebentar (R9)

Berdasarkan Tabel 3. Fuzzy Rule diatas, dapat dijabarkan rule yang akan digunakan dalam penggerjaan proyek akhir ini , diantaranya:

- (R1) IF kelembaban is Kering AND Suhu is Dingin THEN Timer is Lama
- (R2) IF kelembaban is Kering AND Suhu is Normal THEN Timer is Lama
- (R3) IF kelembaban is Kering AND Suhu is Panas THEN Timer is Lama
- (R4) IF kelembaban is Normal AND Suhu is Dingin THEN Timer is Sebentar
- (R5) IF kelembaban is Normal AND Suhu is Normal THEN Timer is Normal
- (R6) IF kelembaban is Normal AND Suhu is Panas THEN Timer is Normal
- (R7) IF kelembaban is Basah AND Suhu is Dingin THEN Timer is Sebentar

(R8) IF kelembaban is Basah AND Suhu is Normal THEN Timer is Sebentar

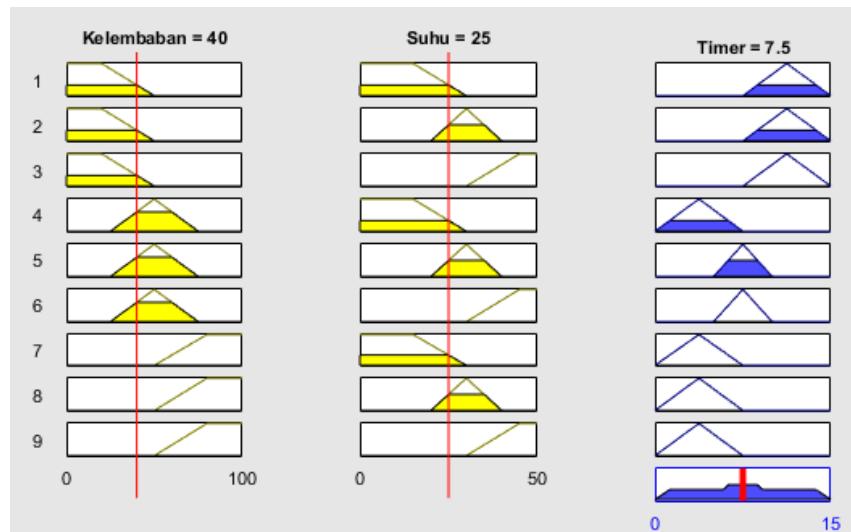
(R9) IF kelembaban is Basah AND Suhu is Panas THEN Timer is Sebentar

Setelah basis rule telah disusun sedemikian rupa, kemudian menentukan fungsi implikasi dari tiap rule, pada pengajaran proyek akhir ini menggunakan metode mamdani dalam membuat sistem cerdas logika fuzzy. Aturan pada metode mamdani untuk fungsi implikasi adalah dengan menggunakan fungsi MIN dengan AND sebagai operator untuk menentukan nilai keanggotaan terkecil, adapun ilustrasi dari persamaan fungsi MIN dengan operator AND:

$$\mu_{A \cap B} = \min(\mu_A[x], \mu_B[y])$$

Contoh kasus untuk dapat memahami subrutin dari program rule fuzzy:

Asumsikan bahwa dari fuzzyifikasi sebelumnya, suhu = 25°C dan kelembaban tanah = 40, adapun dapat diilustrasikan dalam matlab akan tampak seperti Gambar 36:



**Gambar 36. Ilustrasi Hasil berdasarkan Fuzzy Rule**

Pada gambar di atas dapat dilihat bahwa rule yang memenuhi dari fuzzyifikasi suhu suhu = 25°C dan kelembaban tanah = 40 adalah Rule nomor 1, 2, 4 dan 5. Maka bila diimplementasikan secara perhitungan manual kelima rule yang kosong dapat diabaikan sehingga penulisan dari ke4 rule yang memenuhi syarat adalah sebagai berikut:

(R1) IF kelembaban is Kering AND Suhu is Dingin THEN Timer is Lama

$$\begin{aligned}
 \alpha - \text{predikat 1} &= \mu_{\text{Kelembaban tanah}} \cap \mu_{\text{Suhu}} \\
 &= \min(\mu_{\text{Kering}}[40], \mu_{\text{Dingin}}[25]) \\
 &= \min(0.3; 0.3) \\
 &= 0.3
 \end{aligned}$$

(R2) IF kelembaban is Kering AND Suhu is Normal THEN Timer is Lama

$$\begin{aligned}
 \alpha - \text{predikat 2} &= \mu_{\text{Kelembaban tanah}} \cap \mu_{\text{Suhu}} \\
 &= \min(\mu_{\text{Kering}}[40], \mu_{\text{Normal}}[25]) \\
 &= \min(0.3; 0.5) \\
 &= 0.3
 \end{aligned}$$

(R4) IF kelembaban is Normal AND Suhu is Dingin THEN Timer is Sebentar

$$\begin{aligned}
 \alpha - \text{predikat 4} &= \mu_{\text{Kelembaban tanah}} \cap \mu_{\text{Suhu}} \\
 &= \min(\mu_{\text{Normal}}[40], \mu_{\text{Dingin}}[25])
 \end{aligned}$$

$$= \min(0.6; 0.33)$$

$$= 0.33$$

(R5) IF kelembaban is Normal AND Suhu is Normal THEN Timer is Normal

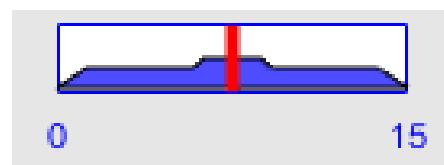
$$\alpha - \text{predikat 5} = \mu_{\text{Kelembaban tanah}} \cap \mu_{\text{Suhu}}$$

$$= \min(\mu_{\text{Normal}}[40], \mu_{\text{Normal}}[25])$$

$$= \min(0.6; 0.5)$$

$$= 0.5$$

Setelah hasil perhitungan dari tiap rule diatas dengan memenuhi kriteria MINIMAL, langkah selanjutnya adalah menkomposisikan tiap rule diatas menjadi satu dengan menggunakan metode MAX, hasilnya dapat diilustrasikan seperti pada gambar 37.



**Gambar 37. Hasil Komposisi Fuzzy Rule**

### 3) Defuzzyfikasi

Defuzzyfikasi adalah suatu proses penegasan ulang tiap himpunan fuzzy yang didapat dari komposisi aturan-aturan fuzzy, setelah komposisi tiap aturan ditegaskan ulang kemudian akan didapatkan suatu output yang merupakan suatu

bilangan pada domain himpunan fuzzy tersebut. Contoh jika diberikan himpunan fuzzy dalam range tertentu, maka dapat diambil dari range tersebut suatu nilai *crisp* sebagai output. Pada proyek akhir ini metode defuzzyifikasi yang digunakan adalah menggunakan metode centroid atau *Center of Area* (COA). Alasan digunakannya metode centroid karena cara pengambilan titik tengah dari tiap daerah fuzzy.

Pada metode centroid untuk penegasan ulang defuzzyifikasi yang artinya titik tengah dari setiap fungsi keanggotaan dapat dicari terlebih dahulu, untuk mencari nilai tengah dari fungsi keanggotaan Sebentra, Normal, dan Lama, digunakan persamaan luas segitiga karena ketiganya mempunyai bentuk segitiga pada fungsi keanggotaannya. Contoh perhitungan dapat kita ilustrasikan seperti berikut ini:

Pada fungsi keanggotaan “Sebentar”, untuk mencari nilai tengahnya dapat dihitung luas daerah dengan cara mengalikan alas dengan tinggi kemudian dibagi 2 setelah itu ditambahkan dengan range nilai minimum dari fungsi keanggotaan Sebentar yakni 0,

$$\text{Panjang} = 7.5 - 0 = 7.5$$

$$\text{Tinggi} = 1$$

$$\begin{aligned}\text{Luas daerah} &= \frac{a \times t}{2} \\ &= \frac{7.5 \times 1}{2} = 3.75\end{aligned}$$

$$\text{Nilai tengah} = 0 + 3.75 = 3.75$$

Untuk fungsi keanggotaan dari normal dan lama, dapat dilakukan perhitungan dengan cara yang sama yaitu mencari luas daerah dan titik tengah sehingga dapat diilustrasikan sebagai berikut:

Normal,

$$\text{Panjang} = 10 - 5 = 5$$

$$\text{Tinggi} = 1$$

$$\text{Luas daerah} = \frac{a \times t}{2}$$

$$= \frac{5 \times 1}{2} = 2.5$$

$$\text{Nilai tengah} = 5 + 2.5 = 7.5$$

Lama,

$$\text{Panjang} = 15 - 7.5 = 7.5$$

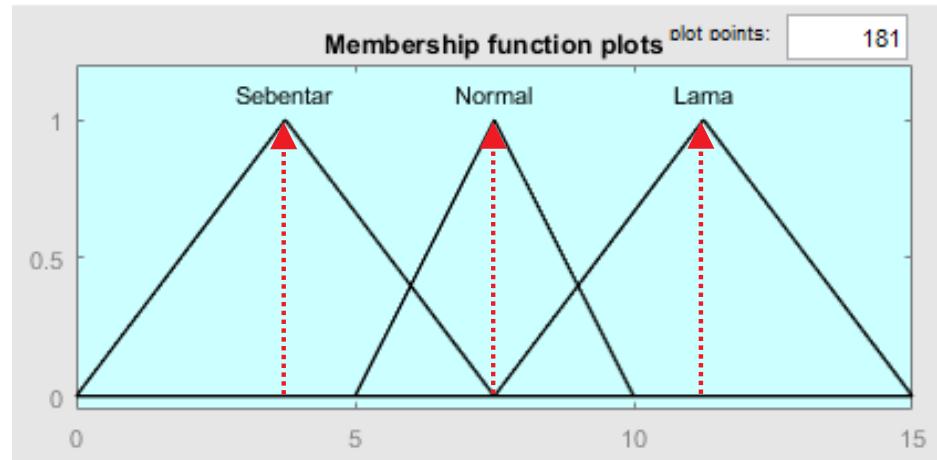
$$\text{Tinggi} = 1$$

$$\text{Luas daerah} = \frac{a \times t}{2}$$

$$= \frac{7.5 \times 1}{2} = 3.75$$

$$\text{Nilai tengah} = 7.5 + 3.75 = 11.25$$

Secara keseluruhan dapat diilustrasikan nilai tengah dari tiap fungsi keanggotaan variabel waktu (*timer*) pada Matlab dapat dilihat pada gambar 38.



**Gambar 38. Nilai Tengah dari Setiap Fungsi Keanggotaan Variabel Waktu (*timer*)**

**Tabel 4. Data Luas dan Titik Tengah**

Anggota	Luas		Titik Tengah (CoA)	
Sebentar	Z1	3.75	A1	3.75
Normal	Z2	2.5	A2	7.5
Lama	Z3	3.75	A3	11.25

Setelah Data Luas dan Titik Tengah diperoleh seperti pada Tabel 4, langkah selanjutnya adalah menentukan nilai dari output yang diharapkan menggunakan persamaan sebagai berikut:

$$Output = \frac{\sum_{i=1}^n rule(i) \times CoA(i) \times luas(i)}{\sum_{i=1}^n rule(i) \times luas(i)}$$

Keterangan :

Rule = Fungsi implikasi fuzzyfikasi

CoA = Titik tengah fungsi keanggotaan variabel waktu (*timer*)

Luas = Luas daerah keanggotaan variabel waktu (*timer*)

n = Jumlah rule yang digunakan

i = data ke-n

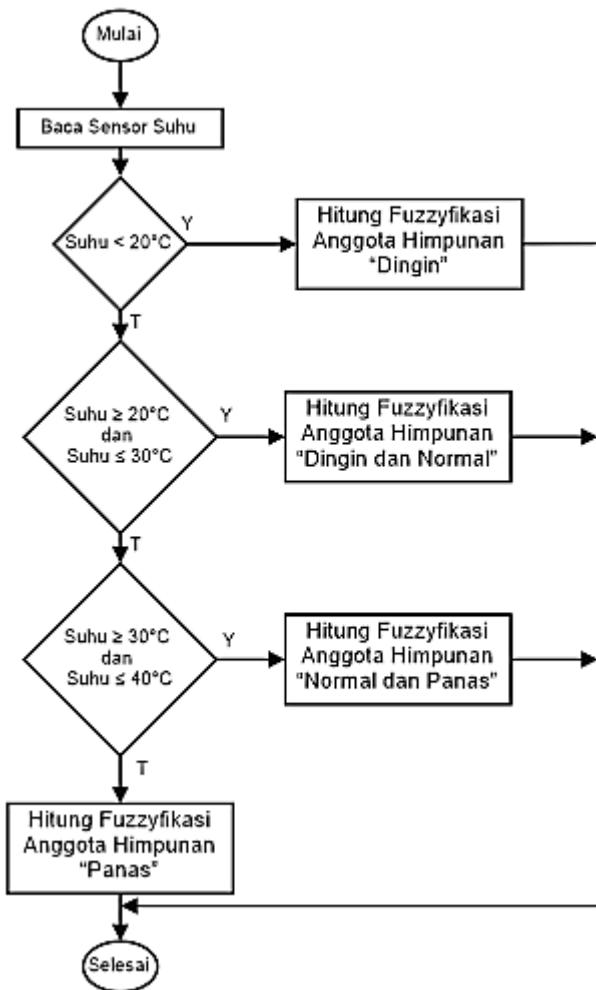
### b. Pemrograman Mikrokontroller

Pada pemrograman mikrokontroller digunakan *software* Arduino IDE sebagai *complier* bahasa C yang digunakan untuk IC ATMega 328.

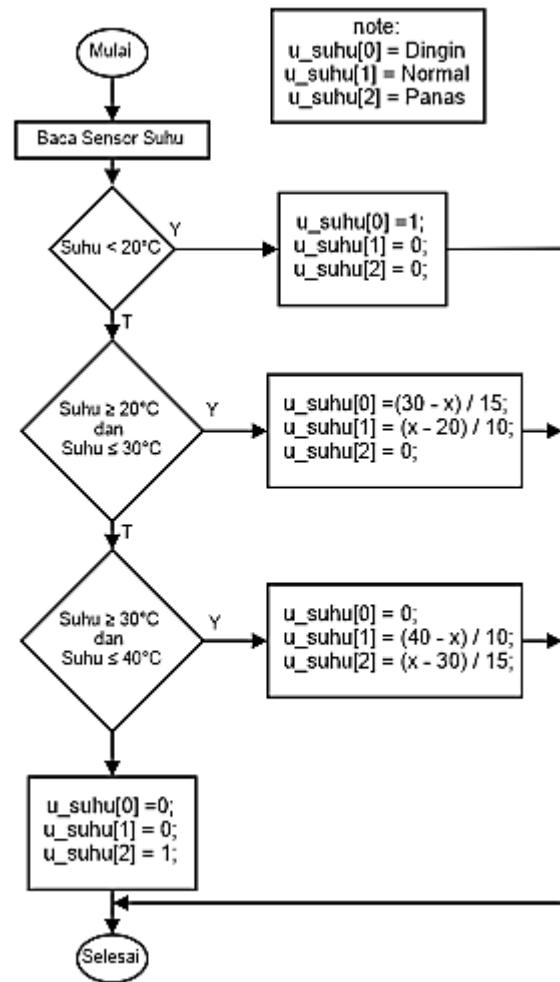
#### 1) Fuzzyfikasi

- Fuzzyfikasi Suhu

Gambar 39 dan gambar 40 merupakan flowchart algoritma dan program fuzzyfikasi variabel suhu yang merupakan implementasi dari Matlab dan kemudian di *complier* ke dalam bahasa C.



**Gambar 39. Flowchart Algoritma Fuzzyifikasi Variabel Suhu**



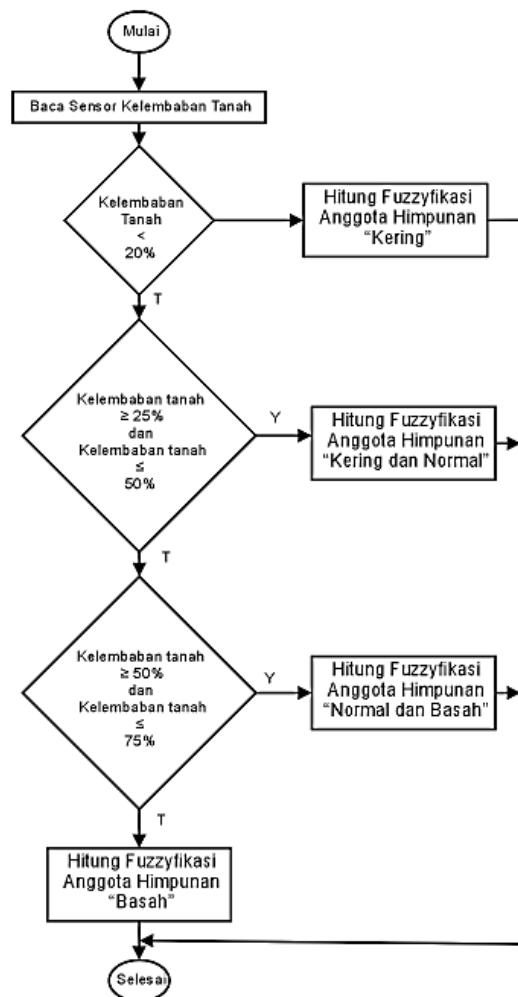
**Gambar 40. Flowchart Program Fuzzyifikasi Variabel Suhu**

Variabel suhu yang diperoleh dari pengambil nilai suhu dari sensor suhu.

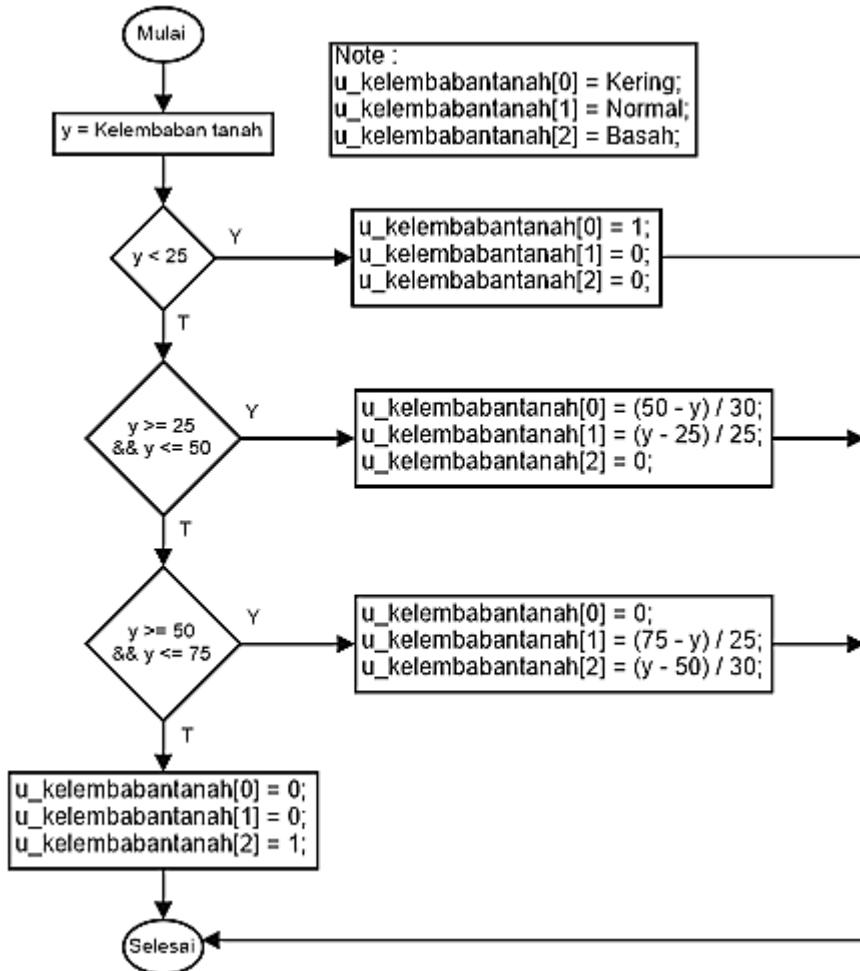
Setalah variabel berhasil disandingkan kemudian variabel suhu dibandingkan dengan kondisi minimal, sedang, dan maksimal yaitu 0, 30, dan 50 untuk menentukan nilai dari derajat keanggotaannya.

- Fuzzyifikasi Kelembaban Tanah

Gambar 41 dan gambar 42 merupakan flowchart algoritma dan program fuzzyifikasi variabel kelembaban tanah yang merupakan implementasi dari Matlab dan kemudian di *complier* ke dalam bahasa C.



**Gambar 41. Flowchart Algoritma Fuzzyifikasi Variabel Kelembaban Tanah**

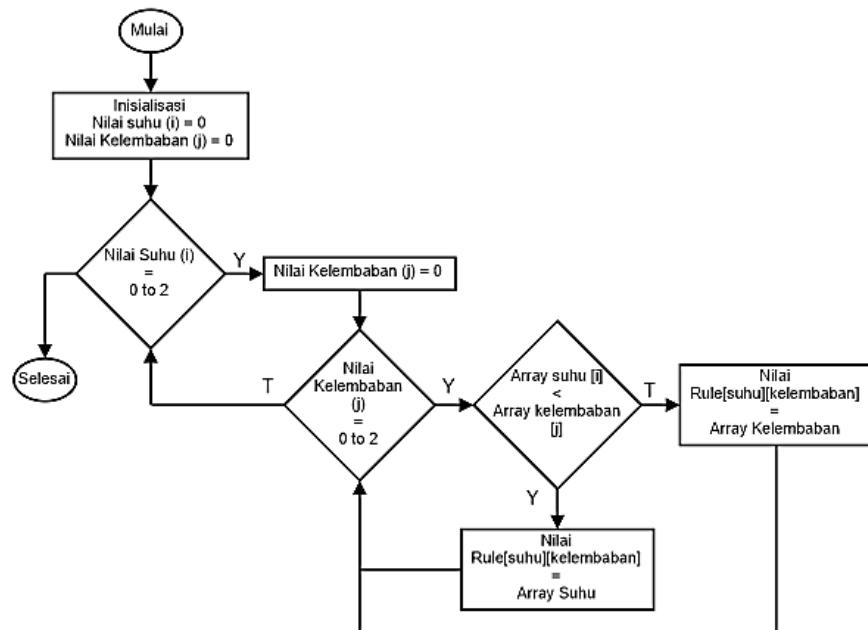


**Gambar 42. Flowchart Program Fuzzyifikasi Variabel Kelembaban Tanah**

Variabel kelembaban tanah yang diperoleh dari pengambil nilai kelembaban tanah dari sensor kelembaban tanah. Setalah variabel berhasil disandingkan kemudian variabel kelembaban tanah dibandingkan dengan kondisi minimal, sedang, dan maksimal yaitu 0, 50, dan 100 untuk menentukan nilai dari derajat keanggotaannya.

## 2) Fuzzy Rule

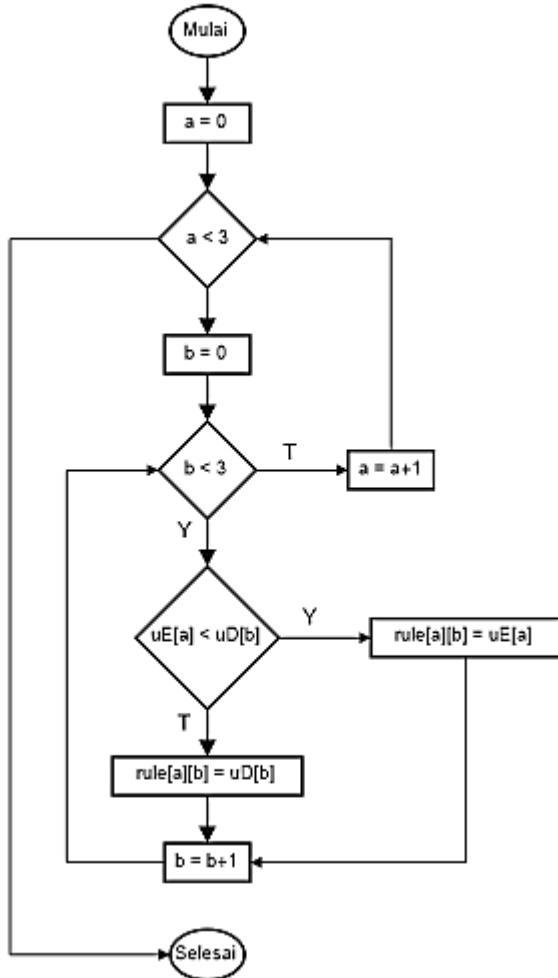
Berikut ini merupakan flowchart dari aturan fuzzy yang akan digunakan pada proyek akhir ini di tunjukan pada Gambar 43 dan Gambar 44 yang di *complier* ke dalam bahasa C.



**Gambar 43. Flowchart Algoritma Fuzzy Rule**

Pada pemrograman menggunakan bahasa C aturan fuzzy di *looping* dengan tujuan untuk memeriksa apakah 9 rule tersebut telah memenuhi syarat dari metode mamdani. Tiap *rule* akan membandingkan nilai derajat keanggotaan dari variabel suhu dan variabel kelembaban tanah, untuk subrutin program diatas *u\_Suhu* merupakan anggota variabel suhu dan *u\_kelembaban* tanah merupakan anggota dari variabel kelembaban tanah, dengan menggunakan MIN maka nilai yang

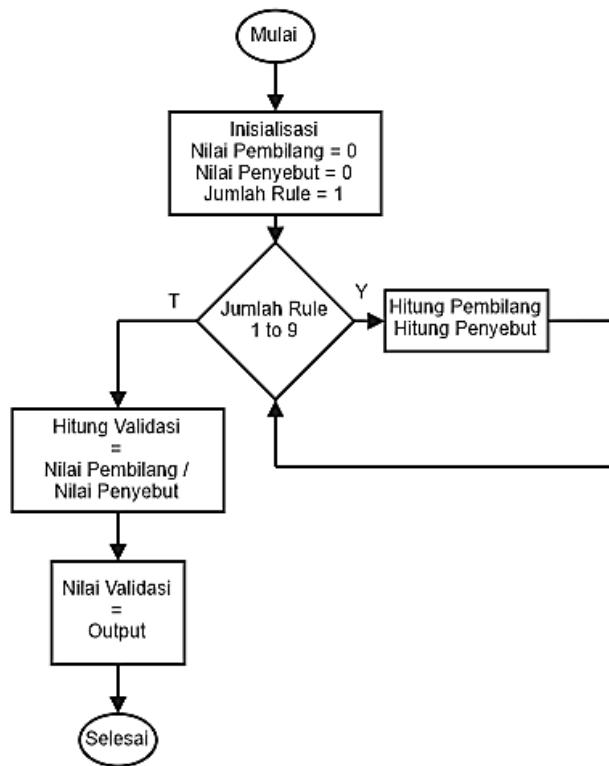
terkecil tiap rule akan diambil kemudian untuk didefuzzyfikasikan agar dapat dilihat keluarannya.



**Gambar 44. Flowchart Program Fuzzy Rule**

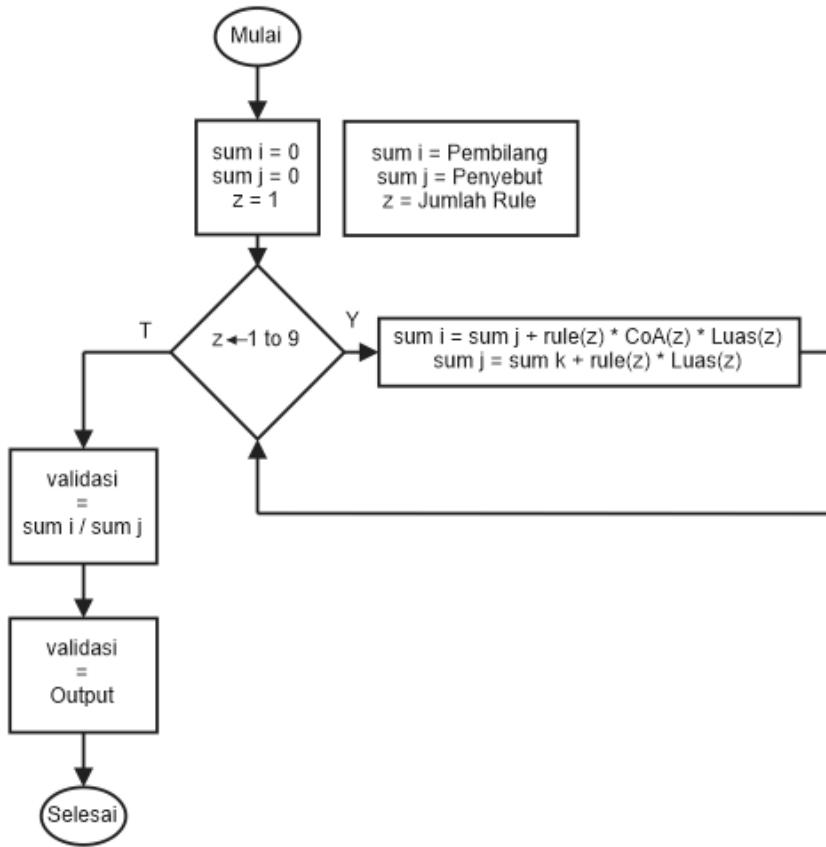
### 3) Defuzzyfikasi

Berikut ini merupakan flowchart dan subrutin program defuzzyfikasi yang digunakan pada proyek akhir ini ditunjukan pada Gambar 45 dan Gambar 46.



**Gambar 45. Flowchart Algoritma Defuzzyifikasi**

Flowchart pada gambar 45 dan gambar didasarkan pada persamaan defuzzyifikasi menggunakan metode *center of area* .

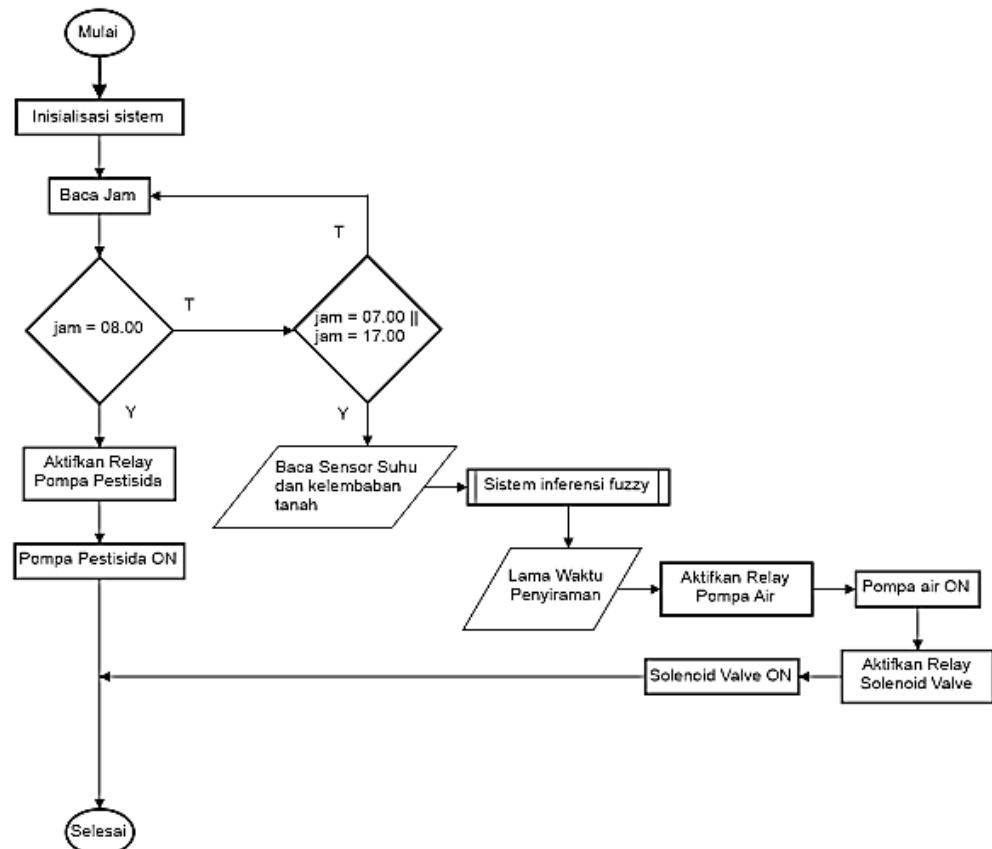


**Gambar 46. Flowchart Program Defuzzyifikasi**

#### 4) Program Alat

Ketika alat ini dijalankan perintah pertama yang dikerjakan adalah pembacaan waktu dari modul *Real Time Clock* oleh mikrokontroller. Ketika waktu yang terbaca oleh mikrokontroller adalah pukul 8 pagi maka akan mengaktifkan relay selama beberapa detik yang mengakibatkan pompa pestisida aktif sesuai dengan relay tersebut. Jika waktu yang ditunjukan adalah pukul 7 pagi atau pukul 5 sore maka perintah yang akan dilakukan yaitu pembacaan sensor suhu dan

kelembaban tanah. Nilai dari sensor suhu dan kelembaban tanah akan menjadi *input* dari sistem inferensi fuzzy. Di dalam subrutin sistem inferensi fuzzy akan diproses sesuai dengan prosedurnya seperti fuzzyifikasi, fuzzy rule, dan defuzzyifikasi. *Output* dari sistem inferensi fuzzy merupakan waktu untuk lamanya penyiraman. Sistem inferensi fuzzy selesai, selanjutnya mengaktifkan relay pompa air sehingga pompa air menyala kemudian mengaktifkan relay solenoid valve sesuai dengan *output* dari sistem inferensi fuzzy sehingga *solenoid valve* terbuka. Berikut ini flowchart program alat dapat dilihat pada gambar 47.



**Gambar 47. Flowchart Algoritma Sistem Keseluruhan**

## C. DEVELOPMENT

*Development* atau pengembangan dalam proyek akhir ini merupakan pengembangan dari perencanaan yang telah dibuat kebentuk *hardware* atau *software* yang siap untuk diimplementasikan. Berikut merupakan tahapan dari proses pengembangan:

### 1. *Hardware*

#### a. Mekanik

- 1) Menyiapkan alat dan bahan yang dibutuhkan dengan menganalisis kebutuhan elektronik.
- 2) Merealisasikan rancang bangun frame alat sesuai dengan rancangan yang dibuat menggunakan besi siku.
- 3) Merealisasikan rancang bangun penyiraman air dan pestisida sesuai dengan rancangan yang dibuat menggunakan pipa paralon dan selang.
- 4) Merealisasikan rancang bangun panel box sesuai dengan rancangan yang dibuat menggunakan plat besi.
- 5) Menguji ketahanan mekanik dari benturan.

#### b. Elektronik

- 1) Menyiapkan alat dan bahan yang dibutuhkan dengan menganalisis kebutuhan elektronik
- 2) Merealisasikan shield mikrokontroller menggunakan PCB dengan desain yang sudah di rancang.
- 3) Menguji unjuk kerja tiap komponen elektronik yang telah dibuat.

## 2. *Software*

*Software* yang digunakan adalah Matlab dan Arduino IDE . *Software* Matlab digunakan untuk memprogram fuzzyifikasi, rule fuzzy, dan defuzzyifikasi. *Software* Arduino IDE digunakan untuk memprogram mikrokontroller. Berikut ini pengembangan Matlab dan Arduino IDE.

### a. Logika Fuzzy (Matlab)

- 1) Menyiapkan alat dan bahan yang dibutuhkan dengan menganalisis kebutuhan baik untuk *input*, proses maupun *output*.
- 2) Mensimulasikan program Matlab yang telah dibuat.

### b. Arduino IDE

- 1) Menyiapkan alat dan bahan yang dibutuhkan dengan menganalisis kebutuhan baik untuk *input*, proses maupun *output*
- 2) Merealisasikan desain *flowchart* ke bentuk program dan mensimulasikan ke bentuk *hardware*.
- 3) Menguji program secara keseluruhan dan menampilkan hasil *output* dari program.

## D. IMPLEMENTATION

*Implementation* atau implementasi merupakan tahapan merealisasikan tahapan pengembangan yang telah dibuat, Berikut merupakan hasil dari tahap implementasi dari perancangan yang terdiri dari desain *hardware* dan *software*.

## 1. *Hardware*

### a. Mekanik

#### 1) Frame Alart

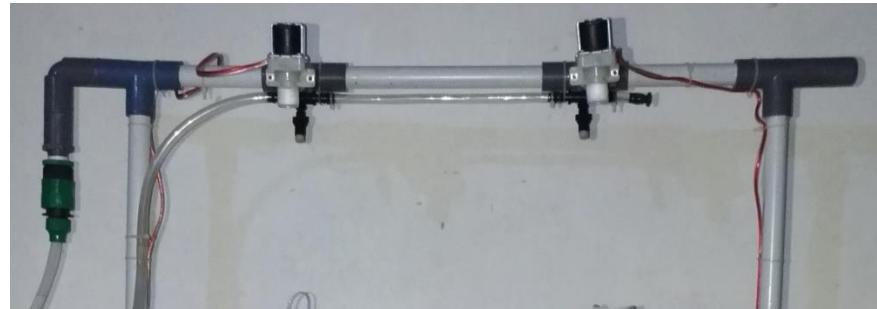


**Gambar 48. Frame Alat**

Gambar 48 merupakan implementasi dari desain frame alat yang telah dipasang semua komponen pendukung. Frame alat adalah bagian yang terpenting dari proyek akhir ini

#### 2) Penyiraman Air dan Pestisida

Gambar 49 merupakan implementasi dari desain penyiraman air dan pestisida yang telah terpasang pada frame alat.



**Gambar 49. Penyiraman air dan pestisida**

3) Panel Box

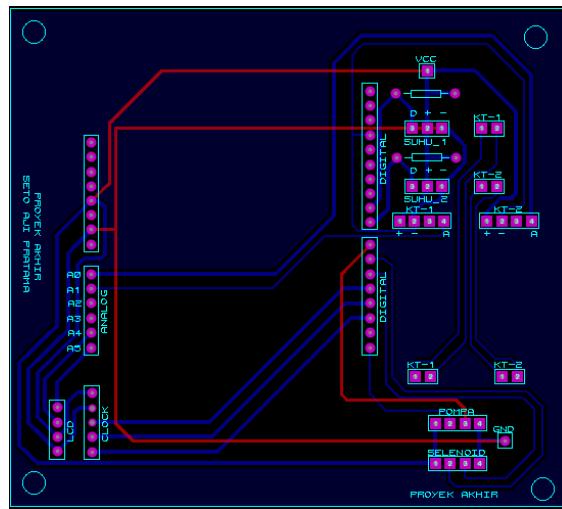


**Gambar 50. Panel Box**

Gambar 50 merupakan implementasi dari desain panel box yang telah terpasang pada frame alat.

b. Elektronik

Gambar 51 merupakan layout PCB shield Mikrokontroller dari Arduino Uno. Desain layout telah disesuaikan dengan kebutuhan I/O yang dibutuhkan agar mempermudah pemasangan perkabelan.



**Gambar 51. Layout PCB Shield Mikrokontroller**

## 2. *Software*

*Software* di realisasikan pada alat proyek akhir ini yaitu dengan cara mengupload program dari Arduino IDE. Lalu alat proyek akhir ini dijalankan sesuai dengan pengoperasian alat. Implementasi pada software menggunakan *trial and error*.

## E. EVALUATION

*Evaluation* merupakan tahapan pengujian dari alat yang telah dibuat, Berikut merupakan hasil dari tahap *evaluaiton* dari proyek akhir ini yang terdiri dari Uji Fungsional dan Uji Kinerja.

## 1. Uji Fungsional

Pengujian fungsional merupakan pengujian setiap bagian dari masing-masing alat yang digunakan pada proyek akhir ini. Tujuan dari pengujian fungsional adalah untuk mengetahui perangkat yang digunakan telah bekerja dengan fungsinya. Berikut adalah jabaran dari uji fungsional:

a. Pengujian *Power supply* dan *Step down*

Pengujian *power supply* merupakan untuk mengetahui *output* tegangan pada datasheet dengan pengukuran langsung. Pengujian *step down* juga untuk mengetahui *output* tegangan pada datasheet dengan pengukuran langsung. Pengujian dilakukan sebanyak 2 kali percobaan, adapun Tabel 5 merupakan Tabel rencana pengujian *power supply* dan *step down*.

**Tabel 5. Rencana Pengujian *Power supply* dan *Step down***

Catu daya	Pengujian ke-	V-out (volt)	V-out terbaca (volt)	Selisih tegangan (volt)	Error (%)
Power supply 12V/5A	1				
	2				
Step down 5V/3A	1				
	2				

b. Pengujian Sensor Suhu

Pengujian sensor suhu untuk mengetahui akurasi dan ketelitian dari sensor suhu yang digunakan. Acuan dari sensor suhu adalah thermometer raksasa, percobaan

pada pengujian sensor suhu dilakukan 10 kali percobaan. Tabel 6 merupakan rencana Tabel pengujian sensor suhu.

**Tabel 6. Rencana Pengujian Sensor Suhu**

No	Sensor Suhu DS18b20 (°C)	Termometer(°C)	Error (%)
1			
2			
3			

c. Pengujian Sensor Kelembaban Tanah

Pengujian sensor kelembaban tanah bertujuan untuk mengetahui kinerja dari sensor kelembaban tanah apakah dapat mendeteksi kadar kelembaban tanah dari objek yang diukur. Pengujian dilakukan dengan melakukan penambahan air dari 30 ml sampai 210 ml, kemudian nilai *output* yang keluar diproses menggunakan persamaan perhitungan kadar kelembaban tanah. Pengujian dilakukan sebanyak 7 kali dapat dilihat pada Tabel 7 yang merupakan rencana tabel pengujian kelembaban tanah.

**Tabel 7. Rencana Pengujian Sensor Kelembaban Tanah**

Pengujian	Volume Air (ml)	Nilai Sensor	Kelembaban Tanah (%)
Sensor Kelembaban Tanah			

d. Pengujian Selenoid Valve

Pengujian solenoid valve untuk mengetahui apakah solenoid valve berjalan sesuai dengan fungsinya. Pengujian dilakukan dengan bantuan relay untuk membuka dan menutup katup. Berikut ini Tabel 8 rencana pengujian solenoid valve .

**Tabel 8. Rencana Pengujian Solenoid Valve**

No	Relay	Tegangan Arduino (5V)	Kondisi Relay	Kondisi Solenoid Valve	Keterangan
1	Relay 1				
2	Relay 2				

e. Pengujian Pompa Air

Pengujian pompa Air untuk mengetahui apakah pompa air berjalan sesuai dengan fungsinya. Pengujian dilakukan dengan bantuan relay untuk memompa air. Berikut ini Tabel 9 rencana pengujian pompa air .

**Tabel 9. Rencana Pengujian Pompa Air**

No	Relay	Tegangan Arduino (5V)	Kondisi Relay	Kondisi Pompa	Keterangan
1	Relay 1				
2	Relay 2				

f. Pengujian Logika Fuzzy

Pengujian Logika fuzzy untuk mengetahui apakah sistem logika fuzzy berjalan sesuai dengan perancangan. Pengujian logika dilakukan dengan membandingkan nilai *output* pada Matlab dengan nilai *output* program Arduino uno secara manual. Pengujian dilakukan sebanyak 15 kali percobaan, berikut Tabel 10 merupakan rancangan pengujian logika fuzzy.

**Tabel 10. Rencana Pengujian Logika Fuzzy**

No	Input		Output (detik)		Error (%)
	Kelembaban Tanah	Suhu (°C)	Matlab	(Manual)	
1					
2					
3					

## 2. Uji Kinerja

Uji kinerja untuk mengetahui unjuk kerja dari rancang bangun penyiraman air otomatis dan proteksi hama tanaman menggunakan *fuzzy logic control*. Pada proses pengujian unjuk kerja akan diidentifikasi waktu penyiraman air dan pestisida, sistem logika fuzzy sesuai dengan sensor suhu dan sensor kelembaban, serta *output* waktu penyiraman. Hasil *output* waktu penyiraman dari program Arduino uno akan dibandingkan dengan nilai *output* Matlab. Pengujian dilakukan selama 5 hari. Berikut Tabel 11 merupakan rancangan pengujian unjuk kerja penyiraman air dan Tabel 12 merupakan rancangan pengujian unjuk kerja penyiraman pestisida.

**Tabel 11. Rancangan Pengujian Unjuk Kerja Penyiraman Air**

Pengujian	Waktu	No.	Input		Penyiraman Air (detik)	
			Kelembaban Tanah	Suhu (°C)	Arduino	Matlab
Hari ke	07.00	1				
		2				
	17.00	1				
		2				

**Tabel 12. Rancangan Pengujian Unjuk Kerja Penyiraman**

Pengujian	Waktu	No.	Penyiraman Pestisida
Hari Ke-	08.00	1	

## F. SPESIFIKASI ALAT

Pada pembuatan proyek akhir Rancang Bangun Penyiraman Air Otomatis dan Proteksi Hama Tanaman menggunakan *Fuzzy Logic Control* memiliki spesifikasi sebagai berikut:

1. Ukuran Alat : Panjang 63 cm, Lebar 36 cm, Tinggi 80 cm
2. Mikrokontroller : ATMega328
3. Sensor : Sensor Suhu, Sensor Kelembaban Tanah

4. Pompa : Pompa Air 12 volt DC
5. Bahasa Pemrograman : bahasa C dan C++
6. *Software complier* : Matlab dan Arduino IDE
7. Metode yang digunakan : Logika fuzzy
8. Objek yang dideteksi : Suhu dan Kelembaban Tanah

## **G. PENGOPERASIAN ALAT**

Berikut ini langkah-langkah pengoperasian alat proyek akhir Penyiraman Air Otomatis dan Proteksi Hama Tanaman menggunakan *Fuzzy Logic Control* .

1. Memposisikan alat dibawah sinar matahari
2. Menyiapkan dua buah tanaman yang akan disiram
3. Mengisi air pada tabung penampung
4. Mengisi pestisida pada tabung penampung.
5. Mengecek seluruh komponen yang terpasang
6. Memasang kabel *power supply* ke stop kontak 220 volt
7. Menghidupkan saklar pada panel pox

## BAB IV

### PENGUJIAN DAN PEMBAHASAN

#### A. HASIL PENGUJIAN

Pembuatan proyek akhir Rancang Bangun Penyiraman Air Otomatis dan Proteksi Hama Tanaman menggunakan *Fuzzy Logic Control*, dalam pengujiannya terbagi menjadi dua yaitu pengujian fungsional atau uji tiap bagian pada alat proyek akhir, dan pengujian kinerja atau unjuk kerja penyiraman air menggunakan logika fuzzy dan penyiraman pestisida.

Hasil pengujian akan dimasukan ke dalam Tabel, setelah data hasil pengujian telah dimasukan ke dalam tabel maka langkah selanjutnya dengan mencari nilai *error* (kesalahan). Hasil yang telah didapat akan dirangkum dalam persentase nilai *error* yang dapat dilihat pada persamaan 1 dan persamaan 2 untuk persentase rata-rata *error* .

$$\%error = \frac{Nilai Sensor - Nilai Acuan}{Nilai Acuan} \times 100\% \quad (1)$$

$$\%error = \frac{\sum error}{\sum uji coba} \times 100\% \quad (2)$$

## 1. Pengujian Fungsional

Pengujian fungsional merupakan pengujian tiap bagian atau blok pada alat proyek akhir. Pengujian fungsional bertujuan untuk mengetahui alat dan aplikasi yang telah dibuat apakah bekerja sesuai dengan yang diinginkan.

### a. Pengujian *Power Supply* dan *Step Down*

**Tabel 13. Hasil Pengujian *Power Supply* dan *Step down***

Catu daya	Pengujian ke-	V-out (volt)	V-out terbaca (volt)	Selisih tegangan (volt)
Power supply 12V/5A	1	12	12.2	0.2
	2	12	12.2	0.2
Step down 5V/3A	1	5	5.1	0.1
	2	5	5.1	0.1

Data Tabel 13 merupakan hasil dari pengujian *power supply* dan *step down* untuk mengetahui *output*nya. *Output* dari *power supply* digunakan sebagai *input* pompa air 12 volt sedangkan *output* dari *step down* digunakan sebagai *input* mikrokontroller.

b. Pengujian Sensor Suhu

**Tabel 14. Hasil Pengujian Sensor Suhu dengan Termometer**

No	Sensor Suhu DS18b20 (°C)	Termometer(°C)	Error (%)
1	29.15	29	0.51
2	30.36	30	1.2
3	31.25	31	0.8
4	32.33	32	1.03
5	33.05	33	0.15
6	34.23	34	0.67
7	35.14	35	0.4
8	36.23	36	0.63
9	37.59	37	1.03
10	38.66	38	1.73
Rata-rata Error			0.815

Data Tabel 14 merupakan hasil dari pengujian sensor suhu pada proyek akhir ini. Adapun perhitungan persentase *error* dan Rata-rata *error* dari pengukuran suhu menggunakan sensor suhu DS18b20 adalah sebagai berikut:

Perhitungan persentase *error* pada saat sensor DS18b20 bernilai 29.15 °C dan Termometer bernilai 29°C.

$$Error (\%) = \frac{(Nilai\ DS18b20)-(Nilai\ Termometer)}{Nilai\ Termometer} \times 100\%$$

$$Error (\%) = \frac{(29.15)-(29)}{29} \times 100\%$$

$$Error (\%) = \frac{0.15}{29} \times 100\%$$

$$Error (\%) = 0.51$$

Perhitungan rata-rata *error* pada saat melakukan uji coba sensor suhu DS18b20.

$$\text{Rata-rata } Error (\%) = \frac{\sum error}{\sum uji coba}$$

$$\text{Rata-rata } Error (\%) = \frac{8.15}{10}$$

$$\text{Rata-rata } Error (\%) = 0.815 \%$$

c. Pengujian Sensor Kelembaban Tanah

**Tabel 15. Hasil Pengujian Sensor Kelembaban Tanah**

Pengujian	Volume Air (ml)	Nilai Sensor	Kelembaban Tanah (%)
SENSOR KELEMBABAN TANAH	30	772.3	24.5
	60	668	34.7
	90	543.2	46.9
	120	478.76	53.2
	150	412.26	59.7
	180	367.25	64.1
	210	293.6	71.3

Data Tabel 15 merupakan hasil pengujian sensor kelembaban tanah, untuk mendapatkan nilai persentase kelembaban tanah dalam pengujian sensor kelembaban tanah adalah sebagai berikut:

Persentase kelembaban tanah pada saat volume air 120 ml dengan nilai sensor adalah 478.76, nilai 1023 adalah nilai ADC dari sensor kelembaban tanah yang menggunakan 10 bit.

$$\text{Kelembaban Tanah (\%)} = \frac{(1023 - \text{Nilai Sensor})}{10.23}$$

$$\text{Kelembaban Tanah (\%)} = \frac{(1023 - 478.76)}{10.23}$$

$$\text{Kelembaban Tanah (\%)} = 53.2$$

#### d. Pengujian Solenoid Valve

**Tabel 16. Hasil Pengujian Solenoid Valve**

No	Relay	Tegangan Arduino (5V)	Kondisi Relay	Kondisi Solenoid Valve	Keterangan
1	Relay 1	HIGH	NO - NC	ON (Buka)	Benar
		LOW	NC - NO	OFF (Tutup)	Benar
2	Relay 2	HIGH	NO - NC	ON (Buka)	Benar
		LOW	NC - NO	OFF (Tutup)	Benar

Tabel 16 merupakan hasil pengujian solenoid valve dengan menggunakan 2 buah solenoid valve dan dibantu relay untuk membuka dan menutup katub.

e. Pengujian Pompa Air

**Tabel 17. Hasil Pengujian Pompa Air DC**

No	Relay	Tegangan Arduino (5V)	Kondisi Relay	Kondisi Pompa	Keterangan
1	Relay 1	HIGH	NO - NC	ON	Benar
		LOW	NC -NO	OFF	Benar
2	Relay 2	HIGH	NO - NC	ON	Benar
		LOW	NC -NO	OFF	Benar

Tabel 17 merupakan hasil pengujian pompa air dengan menggunakan 2 buah pompa air dan dibantu relay untuk membuka dan menutup pompa air.

f. Pengujian Logika Fuzzy

**Tabel 18. Pengujian Logika Fuzzy**

No	Input		Output (detik)		Error (%)
	Kelembaban Tanah	Suhu (°C)	Matlab	(Manual)	
1	100	20	3.75	3.75	0
2	80	25	3.75	3.75	0

3	60	27	5.56	6.13	10.25
4	40	29	8.80	8.84	0.45
5	20	31	11.30	11.25	0.4
6	100	33	3.75	3.75	0
7	80	34	3.75	3.75	0
8	60	36	3.75	3.75	0
9	40	38	9.52	9.33	1.9
10	20	40	11.3	11.25	0.4
11	100	42	3.75	3.75	0
12	80	44	3.75	3.75	0
13	60	46	5.56	5.89	0.9
14	40	48	9.44	9.10	3.6
15	20	50	11.3	11.25	0.4
Rata-rata Error					1.53

Data Tabel 18 merupakan pengujian logika fuzzy ,adapun perhitungan secara manual dengan cara memasukan nilai *input* kelembaban tanah dan suhu ke dalam program arduino, kemudian hitung secara manual.

Contoh input kelembaban tanah 40% dan suhu 29°C, maka outputnya adalah sebagai berikut:

Langkah I : Cari nilai Fuzzyifikasi tiap input

Dengan menggunakan rumus fuzzyifikasi yang terdapat dalam program arduino maka didapatkan nilai fuzzyifikasi tiap input.

Suhu [0] = 0.067 (Dingin)

Kelembaban Tanah [0] = 0.3 (Kering)

Suhu [1] = 0.9 (Normal)

Kelembaban Tanah [1] = 0.6 (Normal)

Suhu [2] = 0 (Panas)

Kelembaban Tanah [2] = 0 (Basah)

Langkah II : Membandingkan nilai fuzzyifikasi tiap input (Rule Fuzzy)

Untuk menentukan nilai tiap rule, dilakukan dengan cara membandingkan nilai fuzzyifikasi tiap input dan cari nilai yang terkecil dapat dilihat pada tabel 17

**Tabel 19. Perhitungan Fuzzy Rule**

Suhu	Kelembaban	Rule	No. Rule
0	0	[0] [0] = 0.067	1
	1	[0] [1] = 0.067	2
	2	[0] [2] = 0	3
1	0	[1] [0] = 0.3	4
	1	[1] [1] = 0.6	5
	2	[1] [2] = 0	6
2	0	[2] [0] = 0	7
	1	[2] [1] = 0	8
	2	[2] [1] = 0	9

Langkah III : Menghitung Defuzzyifikasi

Setelah nilai tiap rule diperoleh, langkah selanjutnya yaitu menghitung defuzzyifikasi dengan menggunakan metode CoA (*Center Of Area*). Metode ini dilakukan dengan cara mengalikan tiap rule dengan luas dan titik tengah dari fungsi keanggotaan variabel waktu (*timer*). Adapun rumus defuzzyifikasi dengan metode CoA (*Center Of Area*) adalah sebagai berikut:

$$Defuzzyifikasi (CoA) = \frac{\sum_{i=1}^n rule(i) \times CoA(i) \times luas(i)}{\sum_{i=1}^n rule(i) \times luas(i)}$$

Untuk nilai Luas dan Titik tengah dari fungsi keanggotaan variabel waktu (*timer*) dapat dilihat pada Bab III. Jadi nilai defuzzyifikasinya adalah

*Defuzzyifikasi (CoA)*

$$\begin{aligned}
 & \frac{\text{rule [0][0]} * L_{Lama} * TT_{Lama} + \text{rule [0][1]} * L_{Sebenatar} * TT_{Sebenatar} +}{\text{rule [1][0]} * L_{Lama} * TT_{Lama} + \text{rule [1][1]} * L_{Normal} * TT_{Normal}} + \\
 & \frac{\text{rule [0][0]} * L_{Lama} + \text{rule [0][1]} * L_{Sebenatar} +}{\text{rule [1][0]} * L_{Lama} + \text{rule [1][1]} * L_{Normal}} \\
 & = \frac{0.067 * 3.75 * 11.25 + 0.067 * 3.75 * 3.75 +}{0.3 * 3.75 * 11.25 + 0.6 * 2.5 * 7.5} + \\
 & \frac{0.067 * 3.75 + 0.067 * 3.75 +}{0.3 * 3.75 + 0.6 * 2.5} \\
 & = \frac{27.6675}{3.1275} \\
 & = 8.84
 \end{aligned}$$

Jadi, nilai defuzzyifikasi dari input kelembaban tanah 40% dan suhu 29°C adalah 8.84 detik.

## 2. Pengujian Kinerja

### a. Unjuk Kerja Penyiraman Air dengan Logika Fuzzy

Pengujian unjuk kerja penyiraman air dengan logika fuzzy bertujuan untuk mengetahui unjuk kerja dari sistem logika fuzzy pada penyiraman air. Unjuk kerja tersebut dimodelkan sebagai ketepatan waktu penyiraman atau *output* sistem logika fuzzy pada mikrokontroller.

Hasil pengujian akan dimasukan ke dalam tabel 20 sesuai dengan hasil percobaan yang dilakukan. Pengujian dilakukan selama 5 hari dengan waktu pagi pukul 07.00 WIB dan sore pukul 17.00 WIB

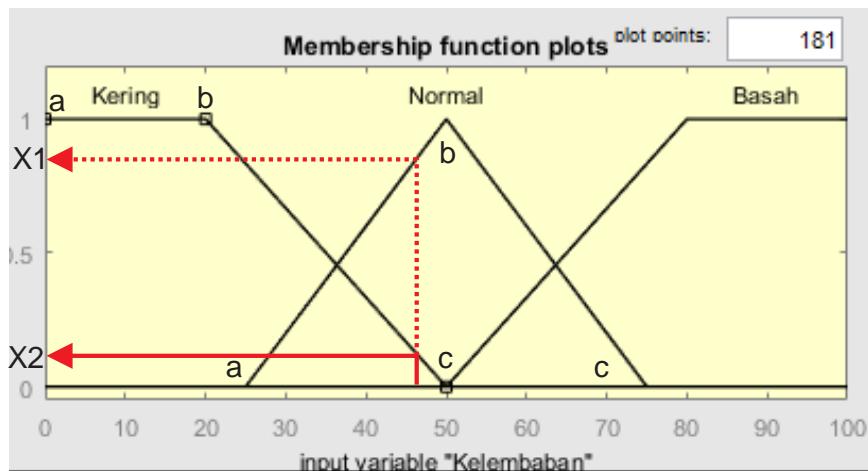
**Tabel 20. Hasil Unjuk Kerja Penyiraman Air dengan Logika Fuzzy**

Pengujian	Waktu	No.	Input		Penyiraman Air (detik)	
			Kelembaban Tanah	Suhu (°C)	Arduino	Matlab
Hari ke-1	07.00	1	66.3	25.3	3.67	4.84
		2	66.5	26.4	3.72	4.8
	17.00	1	57.3	27.5	4.42	5.93
		2	58.1	26.3	4.80	5.77
Hari ke-2	07.00	1	67.1	23.6	3.64	4.89
		2	67.5	23.6	3.62	4.85
	17.00	1	49.4	26.6	5.94	6.14
		2	49.8	26.2	5.59	6.01
Hari ke-3	07.00	1	64.2	22.2	3.46	4.57
		2	64.5	22.2	3.46	4.56
	17.00	1	54.6	27.3	5	6.2
		2	54.6	27.5	5.05	6.28
Hari ke-4	07.00	1	66.8	21.8	3.41	4.39
		2	66.8	20.6	3.16	3.99
	17.00	1	58.2	27.6	4.72	5.8
		2	58.2	27.6	4.72	5.8
Hari ke-5	07.00	1	63.7	23.8	3.69	4.06
		2	64.1	23.5	3.64	4.02
	17.00	1	62.1	25.3	4	5.31
		2	62.5	26.3	4.07	5.26

Adapun contoh perhitungan dari sistem fuzzy pada arduino yang mengacu pada Tabel 20 adalah sebagai berikut :

Misal dari pengujian hari ke-2 pada waktu jam 5 sore didapatkan bahwa input kelembaban tanah sebesar 49.4% serta suhu sebesar 26.6°C, maka didapatkan nilai keanggotaan kelembaban tanah dan suhu.

Nilai keanggotaan kelembaban tanah dapat dilihat pada Gambar 52



**Gambar 52. Membership Function Kelembaban tanah**

Karena nilai  $x_1$  berada diantara  $a$  dan  $b$  dari membership Segitiga (Normal)

$$\text{maka nilai } x_1 = \frac{x-a}{b-a}$$

$$x_1 = \frac{49.4 - 25}{50 - 25}$$

$$x_1 = \frac{24.4}{25}$$

$$x_1 = 0.97$$

dan juga nilai  $x_2$  berada diantara  $b$  dan  $c$  dari membership Trapesium (Kering)

$$\text{maka nilai } x_2 = \frac{c-x}{c-b}$$

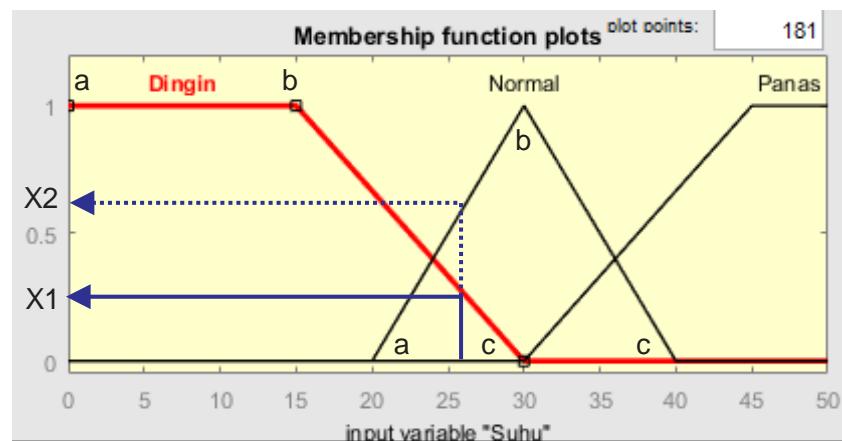
$$x_2 = \frac{50-49.4}{80-50}$$

$$x_2 = \frac{0.6}{30}$$

$$x_2 = 0.02$$

Dari perhitungan diatas dapat dilihat bahwa kelembaban tanah 49.4% mempunyai anggota himpunan normal dengan nilai keanggotaan = 0.976 dan juga anggota himpunan Kering dengan nilai keanggotaan = 0.02. namun kelembaban tanah 49.4% tidak mempunyai anggota himpunan Basah karena nilai keanggotaannya = 0.

Nilai keanggotaan suhu dapat dilihat pada Gambar 53.



**Gambar 53. Membership Function Suhu**

Karena nilai  $x_1$  berada diantara  $b$  dan  $c$  dari membership Trapesium (Dingin) maka nilai  $x_1 = \frac{c-x}{c-b}$

$$x_1 = \frac{30-26.6}{30-15}$$

$$x_1 = \frac{3.4}{15}$$

$$x_1 = 0.22667$$

dan juga nilai  $x_2$  berada di antara  $a$  dan  $b$  dari membership Segitiga (Normal) maka

$$\text{nilai } x_2 = \frac{x-a}{b-a}$$

$$x_2 = \frac{26.6-20}{30-20}$$

$$x_2 = \frac{6.6}{10}$$

$$x_2 = 0.66$$

Dari perhitungan diatas dapat dilihat bahwa suhu  $26.6^{\circ}\text{C}$  mempunyai anggota himpunan dingin dengan nilai keanggotaan = 0.2267 dan juga merupakan anggota himpunan normal dengan nilai keanggotaan = 0.66. namun suhu  $26.6^{\circ}\text{C}$  tidak mempunyai anggota himpunan panas karena nilai keanggotaannya = 0.

Dari proses perhitungan defuzzyfikasi didapatkan bahwa,

Suhu [0] = 0.2267 (Dingin)

Kelembaban Tanah [0] = 0.02 (Kering)

Suhu [1] = 0.66 (Normal)

Kelembaban Tanah [1] = 0.97 (Normal)

Suhu [2] = 0 (Panas)

Kelembaban Tanah [2] = 0 (Basah)

Setelah proses fuzzyfikasi selesai dan menghasilkan nilai keanggotaan tiap input, maka selanjutnya proses fuzzy rule. Pada proses fuzzy rule menggunakan metode MAX-MIN, dimana nilai keanggotaan dari tiap input dibandingkan serta diambil nilai yang terkecil dapat dilihat pada tabel 21.

**Tabel 21. Perhitungan Fuzzy Rule**

Suhu	Kelembaban	Rule	No. Rule
0	0	$[0][0] = 0.02$	1
	1	$[0][1] = 0.2267$	2
	2	$[0][2] = 0$	3
1	0	$[1][0] = 0.02$	4
	1	$[1][1] = 0.66$	5
	2	$[1][2] = 0$	6
2	0	$[2][0] = 0$	7
	1	$[2][1] = 0$	8
	2	$[2][1] = 0$	9

Kemudian masuk ke proses defuzzyifikasi yang menggunakan metode CoA (*Center of Area*).

$$\text{Defuzzyifikasi (CoA)} = \frac{\sum_{i=1}^n \text{rule}(i) \times \text{CoA}(i) \times \text{luas}(i)}{\sum_{i=1}^n \text{rule}(i) \times \text{luas}(i)}$$

Untuk nilai Luas dan Titik tengah dari fungsi keanggotaan variabel waktu (*timer*) dapat dilihat pada Bab III. Jadi nilai defuzzyifikasinya adalah

*Defuzzyifikasi (CoA)*

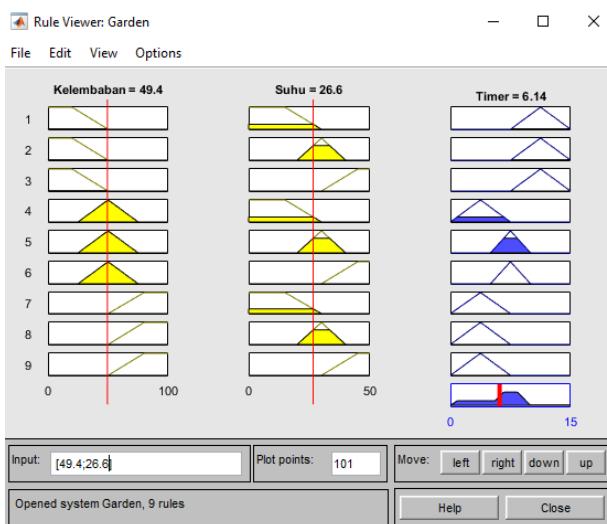
$$\begin{aligned}
 & \text{rule}[0][0] * L_{Lama} * TT_{Lama} + \text{rule}[0][1] * L_{Sebenatar} * TT_{Sebentar} + \\
 & = \frac{\text{rule}[1][0] * L_{Lama} * TT_{Lama} + \text{rule}[1][1] * L_{Normal} * TT_{Normal}}{\text{rule}[0][0] * L_{Lama} + \text{rule}[0][1] * L_{Sebenatar} + \\
 & \quad \text{rule}[1][0] * L_{Lama} + \text{rule}[1][1] * L_{Normal}} \\
 & = \frac{0.02 * 3.75 * 11.25 + 0.2267 * 3.75 * 3.75 +}{0.02 * 3.75 * 11.25 + 0.66 * 2.5 * 7.5} \\
 & \quad \frac{0.02 * 3.75 + 0.22667 * 3.75 +}{0.02 * 3.75 + 0.66 * 2.5}
 \end{aligned}$$

$$= \frac{17.25}{2.904}$$

$$= 5.94$$

Adapun hasil dari defuzzyifikasi merupakan output dari fuzzy yang akan mengaktifkan relay untuk penyiraman air. Untuk menguji apakah *output* sistem fuzzy pada Arduino sama dengan atau mendekati nilai *output* sistem fuzzy pada Matlab sesuai dengan rule yang berlaku atau tidak, maka dilakukan pengujian perhitungan sistem fuzzy pada Matlab.

Disini digunakan program Matlab untuk dapat mensimulasikan pengujian *rule*. Misal dari pengujian hari ke-2 jam 5 sore dengan input kelembaban tanah 49.4 % dan suhu 26.6°C. Nilai input dimasukan ke dalam Matlab agar dapat diperoleh nilai *output*., berikut Gambar 54 merupakan pembuktian rule dengan matlab.



**Gambar 54. Pembuktian Rule dengan Matlab**

b. Unjuk Kerja Penyiraman Pestisida

**Tabel 22. Hasil Pengujian Unjuk Kerja Penyiraman Pestisida**

Pengujian	Waktu	No.	Penyiraman Pestisida
Hari Ke-1	08.00	1	Berhasil
		2	Berhasil
Hari Ke-2	08.00	1	Berhasil
		2	Berhasil
Hari Ke-3	08.00	1	Berhasil
		2	Berhasil
Hari Ke-4	08.00	1	Berhasil
		2	Berhasil
Hari Ke-5	08.00	1	Berhasil
		2	Berhasil

Data Tabel 22 merupakan hasil dari pengujian unjuk kerja penyiraman pestisida yang dilakukan selama 5 hari pada pagi hari pukul 08.00 WIB

## **B. PEMBAHASAN**

Setelah melakukan pengujian keseluruhan terhadap alat proyek akhir yang telah dibangun. Berikut :

### **1. Pembahasan Fungsional**

a. Pengujian *Power Supply* dan *Step Down*.

Pengujian *power supply* dan *step down* pada Tabel 13 merupakan uji tegangan keluaran yang dihasilkan. Hasil pengujian *power supply* dan *stepdown* bekerja dengan baik dan sesuai dengan yang diharapkan. Tegangan output yang terbaca sesuai dengan kebutuhan, namun terdapat selisih tegangan antara tegangan output yang terbaca dengan tegangan output datasheet. Power supply dan *stepdown* tersebut memenuhi tegangan kerja untuk pompa air DC yaitu sebesar 12 V dan mikrokontroller ATMega 328 sebesar 5 V.

b. Pengujian Sensor Suhu

Dari hasil pengujian sensor suhu dapat dilihat pada Tabel 14 hasil pembacaan suhunya kurang lebih sama dengan alat ukur thermometer sebagai pembandingnya. Berdasarkan hasil pada tabel 12 pengujian sensor suhu DS18b20 memiliki *error* 0.815% dengan demikian sensor suhu DS18b20 bekerja dengan baik dan normal.

c. Pengujian Sensor Kelembaban Tanah

Sensor Kelembaban Tanah secara umum dapat bekerja dengan normal sesuai dengan Tabel 13. Dimana jika sensor kelembaban tanah ditancapkan pada tanah kering maka nilai keluarannya kecil, sedangkan jika sensor kelembaban tanah ditancapkan pada tanah basah maka nilai keluarannya besar.

d. Pengujian Solenoid Valve

Dari pengujian yang telah dilakukan, solenoid dapat bekerja dengan baik dan normal sesuai dengan Tabel 14. Ketika relay mendapat sinyal HIGH dari mikrokontroller, Relay akan membuka, sehingga Solenoid Valve dapat merespon

sinyal tegangan 220 volt dengan ditandai solenoid valve terbuka. Sedangkan ketika relay mendapatkan sinyal LOW dari mikrokontroller maka relay akan menutup dan solenoid valve akan menutup.

e. Pengujian Pompa Air

Pompa air dapat bekerja dengan baik dan normal sesuai dengan Tabel 15. Ketika relay mendapat sinyal HIGH dari mikrokontroller, Relay akan membuka, sehingga Pompa Air DC dapat merespon sinyal tegangan 12 volt dari power supply dengan ditandai bunyi motor pada pompa serta dapat memompa air. Sedangkan ketika relay mendapatkan sinyal LOW dari mikrokontroller maka relay akan menutup dan Pompa Air DC tidak mendapatkan tegangan 12 volt.

f. Pengujian Logika Fuzzy

Pengujian Logika fuzzy pada Tabel 18 berjalan dengan baik dan sesuai dengan sistem yang telah dibangun, adapun terdapat rata-rata *error* sebesar 1.53 %. *Error* tersebut didapat dari perbandingan antara hasil perhitungan manual algoritma pada arduino IDE dengan hasil output Matlab, kemudian dihitung menggunakan persamaan 1 dan 2.

## 2. Pembahasan Kinerja

a. Pengujian Penyiraman Air menggunakan *Fuzzy Logic Control*

Pengujian penyiraman air menggunakan *fuzzy logic control* yang mengacu pada tabel 20 dapat dilihat bahwa terdapat beberapa perbedaan antara nilai *output* dari Arduino dengan *output* dari Matlab. Perbedaan tersebut terjadi karena metode

perhitungan yang dilakukan Arduino dengan Matlab berbeda. Perhitungan pada Arduino menggunakan pendekatan 2 angka dibelakang koma sedangkan Matlab perhitungannya *real*. Pengujian penyiraman air menggunakan *fuzzy logic control* dilakukan pada tanaman selama 5 hari pada pagi hari pukul 07.00 WIB dan sore hari pukul 17.00 dengan algoritma fuzzy yang dibangun pada arduino IDE berjalan dengan baik dan sesuai dengan yang diharapkan.

b. Pengujian Penyiraman Pestisida

Pengujian penyiraman pestisida yang mengacu pada tabel 22 dapat dilihat bahwa penyiraman pestisida bekerja dengan baik ketika jadwal penyiraman pestisida.

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **A. KESIMPULAN**

Dapat diambil kesimpulan dari pengujian dan pembahasan proyek akhir mengenai Rancang Bangun Penyiraman Air Otomatis dan Proteksi Hama Tanaman menggunakan *Fuzzy Logic Control*, yakni:

1. Rancang bangun perangkat keras (*hardware*) penyiraman air otomatis dan proteksi hama tanaman menggunakan *fuzzy logic* control sudah berhasil dilaksanakan. Adapun penggunaan alumunium siku setebal 3mm yang disusun persegi panjang dan terdapat dua ruang yang digunakan sebagai tempat media tanam. Untuk penyiraman air dan pestisida menggunakan pompa air DC yang dihubungkan selang dengan paralon serta solenoid valve. Terdapat panel box berukuran 20 x 12 cm dengan bahan plat almunium yang berisi komponen elektronik seperti *Shield* digunakan sebagai tempat meletakan komponen Arduino Uno, *Real Time Clock*, dan pin yang menghubungkan sensor, relay, powersupply, *stepdown*, dan LCD. Serta terdapat dua buah tabung sebagai tempat penampungan sementara air dan pestisida.
2. Perancangan perangkat lunak (*software*) penyiraman air otomatis dan proteksi hama tanaman menggunakan *fuzzy logic* control telah berhasil dilaksanakan.

Untuk Sistem Penyiraman Air Otomatis dan Proteksi Hama Tanaman menggunakan *Fuzzy Logic Control* dibangun dengan *software* MATLAB dan Arduino IDE. Pada pembuatan *Fuzzy Logic Control* digunakan MATLAB untuk menentukan *fuzzyifikasi*, *rule fuzzy*, dan *defuzzyifikasi*. *Software* Arduino IDE digunakan sebagai pembuatan *source code* program yang menggunakan bahasa C, *source code* program tersebut berfungsi untuk menjalankan mikrokontroller Arduino Uno. Dalam program Arduino IDE terdapat beberapa *library* untuk menjalankan komponen-komponen proyek akhir serta *fuzzyifikasi*, *rule fuzzy*, dan *defuzzyifikasi* yang telah ditentukan terlebih dahulu dengan MATLAB. Adapun *error* antara nilai output dari algoritma program yang dibuat dengan nilai output Matlab sebesar 1,53 %.

3. Unjuk kerja dari Sistem Penyiraman Air Otomatis dan Proteksi Hama Tanaman menggunakan *Fuzzy Logic Control* berdasarkan hasil pengujian yang telah dilaksanakan dapat berfungsi dengan baik dari segi sistem cerdas yang berbasis logika fuzzy maupun sistem penyiraman. Sensor kelembaban tanah bekerja dengan baik dan terdapat *error* pada pembacaan sensor suhu sebesar 0,815 %.

## **B. KETERBATASAN ALAT**

Proyek akhir dengan judul “Rancang Bangun Penyiraman Air Otomatis dan Proteksi Hama Tanaman menggunakan *Fuzzy Logic Control*” memiliki keterbatasan alat, adapun keterbatasannya adalah sebagai berikut:

1. Pengaturan jadwal penyiraman pada sistem ini masih menggunakan pengaturan lewat program.
2. Pada perancangan elektronik port I/O masih kurang efisien dan efektif sehingga terdapat kabel *jumber* disekitaran *shield* mikrokontroller, harapannya dapat mengoptimalkan struktur port I/O yang telah disediakan dari ATMega328.
3. Penyiraman pestisida pada sistem ini otomatis dengan menggunakan pengaturan lewat program tanpa menggunakan sistem logika fuzzy.
4. Parameter utama dalam penyiraman air pada tanaman hanya menggunakan suhu dan kelembaban tanah.

### **C. PENELITIAN LANJUTAN**

Berdasarkan hasil dari proyek akhir tersebut, masih terdapat banyak kekurangan dari proyek akhir ini karena keterbatasan materi, kemampuan dan waktu, sehingga penulis menyarankan untuk melakukan penelitian lanjutan sebagai berikut:

1. Pengaturan jadwal penyiraman masih dilakukan lewat program, oleh karena itu diperlukan antarmuka tambahan seperti *keypad* dan tombol-tombol pendukung yang merupakan perangkat masukan sehingga mengatur jadwal penyiraman sesuai dengan waktu yang ditentukan.

2. Tata letak pada perancangan elektronik mempertimbangkan penempatan port I/O dan aspek kebutuhan komponen elektronik, agar dapat disusun secara efisien dan efektif.
3. Pada proses penyiraman pestisida menggunakan sistem logika fuzzy sehingga akan lebih efektif dan efisien. Kedepannya penambahan sensor-sensor pendukung untuk proses penyiraman pestisida harus diperhatikan.
4. Sistem yang dibuat pada proyek akhir ini masih bersifat mendasar karena hanya suhu dan kelembaban tanah yang diambil sebagai parameter utama dalam penyiraman air pada tanaman, untuk kedepannya akan lebih baik jika parameter tidak hanya suhu dan kelembaban tanah, PH tanah, kelembaban udara, dan sensor-sensor lain yang dapat mengoptimalkan kinerja sistem juga harus dapat dijadikan sebagai parameter utama dalam penyiraman air pada tanaman.

## DAFTAR PUSTAKA

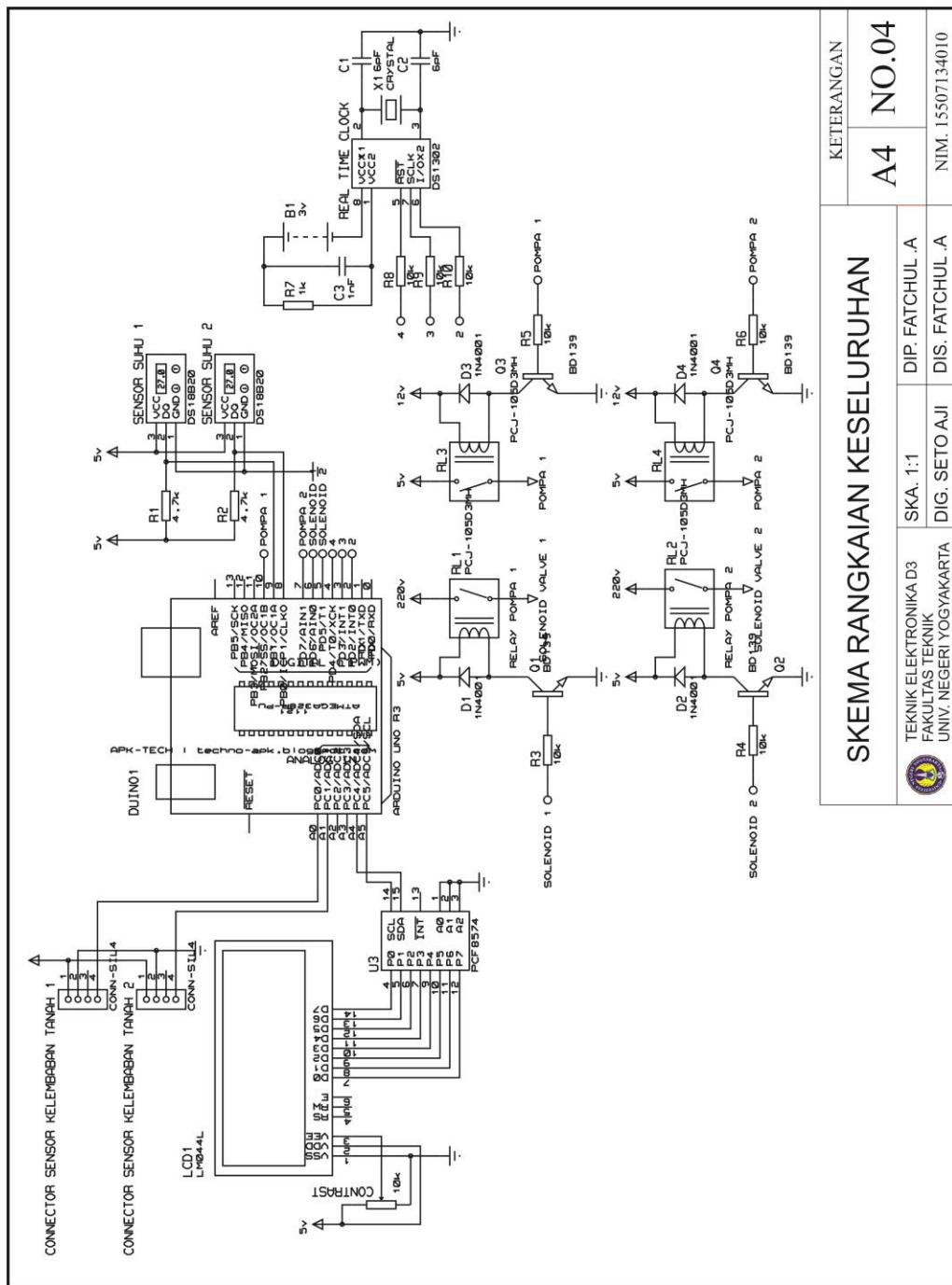
- Budiutomo Nanang. (2018). 8 Hama yang menyerang tanaman cabe serta cara penanganannya. Dari <https://bukubiruku.com/hama-tanaman-cabe/>. Diambil pada tanggal 6 Januari 2018
- Dick dan Carey. (2010). Metode ADDIE. Diambil pada tanggal 6 Januari 2018, dari [http://en.wikipedia.org/wiki/ADDIE\\_Model](http://en.wikipedia.org/wiki/ADDIE_Model)
- Kitoma Indonesia. (2018) . Solenoid Valve pneumatic prinsip kerja. Dari <http://www.kitomaindonesia.com/article/9/solenoid-valve-pneumatic-prinsip-kerja>. Diambil tanggal 4 Januari 2018
- Laboratory Narin. (2018). Sensor Suhu DS18b20 pada Arduino. Dari <https://tutorkeren.com/artikel/tutorial-menggunakan-sensor-suhu-ds18b20-pada-arduino.htm>. Diambil pada tanggal 4 Januari 2018
- Perhorti (2015). Holtikultura. Departemen Agronomi & Hortikultura, Fakultas Pertanian, IPB
- Pracaya (2003). Hama dan Penyakit tanaman. Penebar Swadaya: Depok.
- Pranata Tulus, Beni Irawan dan Ilhamsyah. (2015). Penerapan Logika Fuzzy pada Sistem Penyiraman Tanaman Otomatis Berbasis Mikrokontroller. Jurnal Coding, Sistem Komputer Untan, Volume 03, Nomor 2, hal. 11-12.
- Sri Kusumadewi dan Hari Purnomo. (2013). Aplikasi Logika Fuzzy untuk Pendukung Keputusan. Yogyakarta: Graha Ilmu

- Sri Kusumadewi. (2003). Artificial Intelligence. Yogyakarta: Graha Ilmu
- Wikikomponen. (2018). Prinsip dan Cara Kerja Mesin Pompa Air. Dari <http://www.wikikomponen.com/prinsip-dan-cara-kerja-mesin-pompa-air/>. Diambil pada tanggal 6 Januari 2018
- Suhartono. (2008). Pengaruh Interval Pemberian Air Terhadap Tumbuhan. *Jurnal Embryo*. Vol.5 (1)

# LAMPIRAN

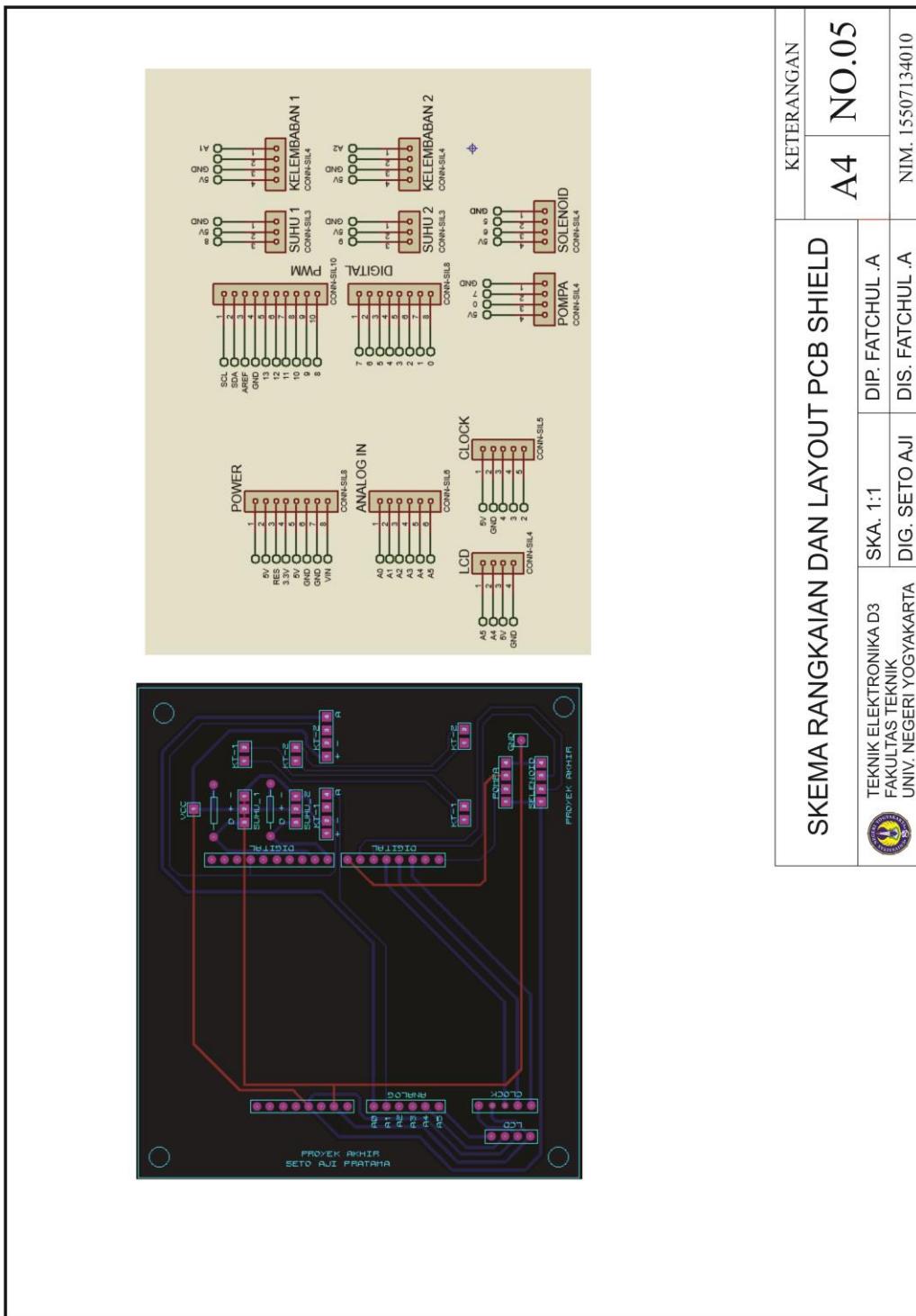
## LAMPIRAN 1

## Skema Rangkaian Keseluruhan



## LAMPIRAN 2

### Skema Rangkaian dan Layout PCB Shield



### LAMPIRAN 3

#### Source Program

```
#include <DallasTemperature.h>
#include <OneWire.h>
#include <Wire.h>
#include <DS1302.h>
#include <LiquidCrystal_I2C.h>
#include "I2Cdev.h"
#if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
    #include "Wire.h"
#endif
#define analogInPin0    A0 // Sensor Kelembaban 1
#define analogInPin1    A1 // Sensor Kelembaban 2
#define ONE_WIRE_BUS_1  9 // Sensor Suhu 1
#define ONE_WIRE_BUS_2  8 // Sensor Suhu 2
#define RELAY_POMPA1    7 // Relay Pompa 1
#define RELAY_POMPA2    10 // Relay Pompa 2
#define RELAY_SELENOID1  5 // Relay Selenoid 1
#define RELAY_SELENOID2  6 // Relay Selenoid 2

LiquidCrystal_I2C lcd (0x3F,2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);
DS1302 rtc(2,3,4);
OneWire oneWire_1(ONE_WIRE_BUS_1);
OneWire oneWire_2(ONE_WIRE_BUS_2);
DallasTemperature sensorSuhu1(&oneWire_1);
DallasTemperature sensorSuhu2(&oneWire_2);

//===== Variabel =====//
```

```

int ValueSensor_1    = 0 ;
int ValueSensor_2    = 0 ;
float suhuSekarang1  = 0 ;
float suhuSekarang2  = 0 ;
float kelembaban_1   = 0 ;
float kelembaban_2   = 0 ;
//=====================================================================

//===================================================================== Rule Base =====
float kelembaban[3];
float suhu[3];
float rule[3][3];
float validasi1, pembilang1, penyebut1, output1;
float validasi2, pembilang2, penyebut2, output2;
//=====================================================================

//===================================================================== Luas & COA =====
float L_Sebentar = 3.75;
float L_Normal   = 2.5;
float L_Lama     = 3.75;
float TT_Sebentar = 3.75;
float TT_Normal  = 7.5;
float TT_Lama    = 11.25;
//=====================================================================

void setup() {
Serial.begin(9600);
rtc.halt(false);
rtc.writeProtect(false);

```

```
lcd.begin(20,4);
pinMode(RELAY_POMPA1,OUTPUT);
pinMode(RELAY_POMPA2,OUTPUT);
pinMode(RELAY_SELENOID1,OUTPUT);
pinMode(RELAY_SELENOID2,OUTPUT);
digitalWrite(RELAY_SELENOID1,HIGH);
digitalWrite(RELAY_SELENOID2,HIGH);
digitalWrite(RELAY_POMPA1,HIGH);
digitalWrite(RELAY_POMPA2,HIGH);
sensorSuhu1.begin();
sensorSuhu2.begin();
greeting();
calibrating();
}

void greeting(){
lcd.setCursor(6,0);
lcd.print("BISMILLAH");
delay(1000);
lcd.setCursor(1,1);
lcd.print("==PROYEK AKHIR==");
lcd.setCursor(0,1);
delay(1000);
lcd.setCursor(2,2);
lcd.print("SETO AJI PRATAMA");
delay(2000);
lcd.clear();
}
```

```
void calibrating(){
    lcd.clear();
    lcd.setCursor(2,1);
    lcd.print("...Please Wait...");
    delay(1000);
    lcd.clear();
}
```

```
void fuzzyfikasi(){
    lcd.clear();
    lcd.setCursor(3,0);
    lcd.print("PROSES FUZZY");
    delay(1000);
    lcd.setCursor(2,1);
    lcd.print("...Please Wait...");
    delay(1000);
    lcd.clear();
}
```

```
void sensor_kelembaban1(){
    ValueSensor_1 = analogRead(analogInPin0);
    //Serial.print("Sensor ADC = ");
    //Serial.print(ValueSensor_1);
    kelembaban_1 = ((1023 - ValueSensor_1)/10.23);
    //Serial.print(", Output Soil Moisture = ");
    //Serial.print(kelembaban_1);
    delay(1000);
}
```

```
void sensor_kelembaban2(){
    ValueSensor_2 = analogRead(analogInPin1);
    //Serial.print("Sensor ADC = ");
    //Serial.print(ValueSensor_1);
    kelembaban_2 =((1023 - ValueSensor_2)/10.23);
    //Serial.print(", Output Soil Moisture = ");
    //Serial.print(kelembaban_1);
    delay(1000);
}
```

```
void sensor_suhu1(){
    suhuSekarang1 = ambilSuhu1();
    delay(500);
}
```

```
float ambilSuhu1(){
    sensorSuhu1.requestTemperatures();
    float suhu1 =sensorSuhu1.getTempCByIndex(0);
    return suhu1;
}
```

```
void sensor_suhu2(){
    suhuSekarang2 = ambilSuhu2();
    delay(500);
}
```

```
float ambilSuhu2(){
    sensorSuhu2.requestTemperatures();
```

```

float suhu2 = sensorSuhu2.getTempCByIndex(0);
return suhu2;
}

void fuzzykelembaban_1(){
if (kelembaban_1 < 25){
    kelembaban[0] = 1;
    kelembaban[1] = 0;
    kelembaban[2] = 0;
}
else if(kelembaban_1 >= 25 && kelembaban_1 <50){
    kelembaban[0] = (50 - kelembaban_1) / 30;
    kelembaban[1] = (kelembaban_1 - 25) / 25;
    kelembaban[2] = 0;
}
else if(kelembaban_1 >= 50 && kelembaban_2 < 75){
    kelembaban[0] = 0;
    kelembaban[1] = (75 - kelembaban_1) / 25;
    kelembaban[2] = (kelembaban_1 - 50) / 30;
}
else{
    kelembaban[0] = 0;
    kelembaban[1] = 0;
    kelembaban[2] = 1;
}
}

```

```

void fuzzykelembaban_2(){
if (kelembaban_2 < 25){
    kelembaban[0] = 1;
    kelembaban[1] = 0;
    kelembaban[2] = 0;
}
else if(kelembaban_2 >= 25 && kelembaban_2 <50){
    kelembaban[0] = (50 - kelembaban_2) / 30;
    kelembaban[1] = (kelembaban_2 - 25) / 25;
    kelembaban[2] = 0;
}
else if(kelembaban_2 >= 50 && kelembaban_2 < 75){
    kelembaban[0] = 0;
    kelembaban[1] = (75 - kelembaban_2) / 25;
    kelembaban[2] = (kelembaban_2 - 50) / 30;
}
else{
    kelembaban[0] = 0;
    kelembaban[1] = 0;
    kelembaban[2] = 1;
}
}

```

```

void fuzzysuhu_1(){
if (suhuSekarang1 < 20){
    suhu[0] = 1;
    suhu[1] = 0;
    suhu[2] = 0;
}

```

```

else if(suhuSekarang1 >= 20 && suhuSekarang1 < 30){
    suhu[0] = (30 - suhuSekarang1) / 15;
    suhu[1] = (suhuSekarang1 - 20) / 10;
    suhu[2] = 0;
}

else if(suhuSekarang1 >= 30 && suhuSekarang1 <= 40){
    suhu[0] = 0;
    suhu[1] = (40 - suhuSekarang1) / 10;
    suhu[2] = (suhuSekarang1 - 30) / 15;
}

else{
    suhu[0] = 0;
    suhu[1] = 0;
    suhu[2] = 1;
}

}

```

```

void fuzzysuhu_2(){
if (suhuSekarang2 < 20){
    suhu[0] = 1;
    suhu[1] = 0;
    suhu[2] = 0;
}

else if(suhuSekarang2 >= 20 && suhuSekarang2 < 30){
    suhu[0] = (30 - suhuSekarang2) / 15;
    suhu[1] = (suhuSekarang2 - 20) / 10;
    suhu[2] = 0;
}

else if(suhuSekarang2 >= 30 && suhuSekarang2 <= 40){

```

```

suhu[0] = 0;
suhu[1] = (40 - suhuSekarang2) / 10;
suhu[2] = (suhuSekarang2 - 30) / 15;
}
else{
    suhu[0] = 0;
    suhu[1] = 0;
    suhu[2] = 1;
}
}

void rule_evaluation(){
    int i,j;
    for(i=0 ; i<3; i=i+1){
        for(j=0; j<3; j=j+1){
            if (suhu[i] < kelembaban[j]){
                rule[i][j] = suhu[i];
            }
            else {
                rule[i][j] = kelembaban[j];
            }
        }
    }
}

//===== Metode Centroid =====/
void defuzzifikasi1(){
    pembilang1 = (rule[0][0] * L_Lama * TT_Lama) + (rule[0][1] * L_Sebentar *
    TT_Sebentar) + (rule[0][2] * L_Sebentar * TT_Sebentar) +

```

```

(rule[1][0] * L_Lama* TT_Lama) + (rule[1][1] * L_Normal * TT_Normal)
+ (rule[1][2] * L_Sebentar * TT_Sebentar) +
    (rule[2][0] * L_Lama * TT_Lama) + (rule[2][1] * L_Normal * TT_Normal)
+ (rule[2][2] * L_Sebentar * TT_Sebentar);

```

```

penyebut1      = (rule[0][0]      * L_Lama)      + (rule[0][1]* L_Sebentar)      +
(rule[0][2]*L_Sebentar) +
    (rule[1][0] * L_Lama) + (rule[1][1] * L_Normal)      + (rule[1][2] *
L_Sebentar) +
    (rule[2][0] * L_Lama) + (rule[2][1] * L_Normal) + (rule[2][2] *
L_Sebentar);

```

```

validasi1 = pembilang1/penyebut1;
output1  = validasi1;
}

```

```

void defuzzifikasi2(){
pembilang2 = (rule[0][0] * L_Lama * TT_Lama) + (rule[0][1] * L_Sebentar *
TT_Sebentar) + (rule[0][2] * L_Sebentar * TT_Sebentar) +
    (rule[1][0] * L_Lama* TT_Lama) + (rule[1][1] * L_Normal * TT_Normal)
+ (rule[1][2] * L_Sebentar * TT_Sebentar) +
    (rule[2][0] * L_Lama * TT_Lama) + (rule[2][1] * L_Normal * TT_Normal)
+ (rule[2][2] * L_Sebentar * TT_Sebentar);

```

```

penyebut2      = (rule[0][0]      * L_Lama)      + (rule[0][1]* L_Sebentar)      +
(rule[0][2]*L_Sebentar) +
    (rule[1][0] * L_Lama) + (rule[1][1] * L_Normal)      + (rule[1][2] *
L_Sebentar) +

```

```

        (rule[2][0] * L_Lama) + (rule[2][1] * L_Normal) + (rule[2][2] *
L_Sebentar);

validasi2 = pembilang2/penyebut2;
output2 = validasi2;
}

void loop() {
    lcd.setCursor(6,0);
    lcd.print(rtc.getTimeStr());
    lcd.setCursor(6,1);
    lcd.print(rtc.getDOWStr(FORMAT_LONG));
    lcd.setCursor(5,2);
    lcd.print(rtc.getDateStr());
    delay(5000);
    lcd.clear();

    sensor_kelembaban1();
    sensor_kelembaban2();
    lcd.setCursor(0,0);
    lcd.print("kelembaban_1:");
    lcd.setCursor(14,0);
    lcd.print(kelembaban_1);
    delay(1000);
    lcd.setCursor(0,1);
    lcd.print("kelembaban_2:");
    lcd.setCursor(14,1);
    lcd.print(kelembaban_2);
    delay(1000);
}

```

```

sensor_suhu1();
sensor_suhu2();
lcd.setCursor(0,2);
lcd.print("Suhu_1:");
lcd.setCursor(9,2);
lcd.print(suhuSekarang1);
delay(1000);
lcd.setCursor(0,3);
lcd.print("Suhu_2:");
lcd.setCursor(9,3);
lcd.print(suhuSekarang2);
delay(1000);
lcd.clear();

if(rtc.getTimeStr() == 7 || rtc.getTimeStr() == 17){
fuzzyfikasi();
fuzzykelembaban_1(); //Fuzzyfikasi
fuzzysuhu_1(); //Fuzzyfikasi
rule_evaluation();
defuzzifikasi1();
Serial.println("validasi1 ");
Serial.println(validasi1);
Serial.println("Hasil1 ");
Serial.println(output1);
delay(1000);
lcd.setCursor(0,0);
lcd.print("Hasil1 : ");
lcd.setCursor(9,0);
lcd.print(output1);

```

```
lcd.setCursor(0,1);

lcd.print("Waktu siram1(s): ");
lcd.setCursor(0,2);
lcd.print(output1);
delay(1000);

digitalWrite(RELAY_POMPA1,LOW);
digitalWrite(RELAY_SELENOID1,LOW);
delay(output1*1000);
digitalWrite(RELAY_POMPA1,HIGH);
digitalWrite(RELAY_SELENOID1,HIGH);

fuzzyfikasi();
fuzzykelembaban_2();
fuzzysuhu_2();
rule_evaluation();
defuzzifikasi2();
Serial.println("validasi2 ");
Serial.println(validasi2);
Serial.println("Hasil2 ");
Serial.println(output2);
delay(1000);
lcd.setCursor(0,0);
lcd.print("Hasil2 : ");
lcd.setCursor(9,0);
lcd.print(output2);
lcd.setCursor(0,1);
lcd.print("Waktu siram2 (s)");
lcd.setCursor(0,2);
```

```
lcd.print(output2);
delay(1000);
digitalWrite(RELAY_POMPA1,LOW);
digitalWrite(RELAY_SELENOID2,LOW);
delay(output2*1000);
digitalWrite(RELAY_POMPA1,HIGH);
digitalWrite(RELAY_SELENOID2,HIGH);
lcd.clear();
}
else if (rtc.getTimeStr() == 8){

lcd.setCursor(0,0);
lcd.print("Penyiraman Pestisida");
lcd.setCursor(0,1);
lcd.print("==== Please Wait ===");

digitalWrite(RELAY_POMPA2,LOW);
delay(2000);
digitalWrite(RELAY_POMPA2,HIGH);
lcd.clear();
}
}
```

## LAMPIRAN 4

### Datasheet ATMega328



8-bit AVR Microcontrollers

ATmega328/P

DATASHEET SUMMARY

#### Introduction

The Atmel® picoPower® ATmega328/P is a low-power CMOS 8-bit microcontroller based on the AVR® enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega328/P achieves throughputs close to 1MIPS per MHz. This empowers system designer to optimize the device for power consumption versus processing speed.

#### Feature

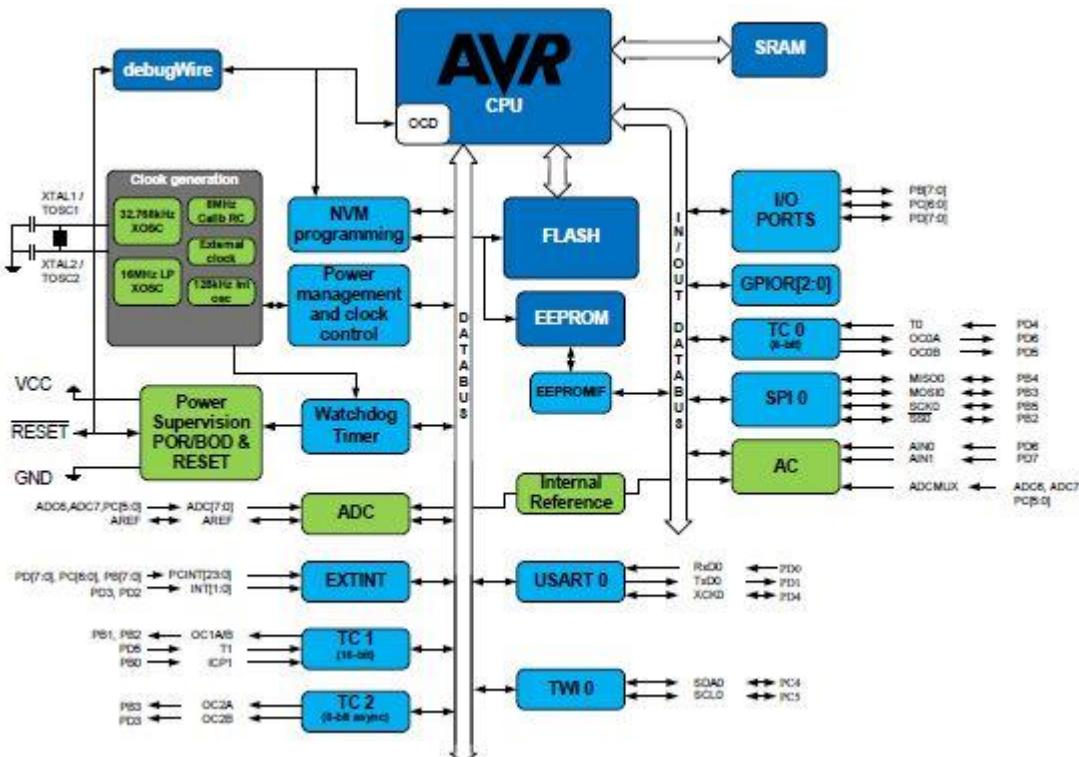
##### High Performance, Low Power Atmel®AVR® 8-Bit Microcontroller Family

- Advanced RISC Architecture
  - 131 Powerful Instructions
  - Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 20 MIPS Throughput at 20MHz
  - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
  - 32KBytes of In-System Self-Programmable Flash program Memory
  - 1KBytes EEPROM
  - 2KBytes Internal SRAM
  - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
  - Data Retention: 20 years at 85°C/100 years at 25°C<sup>(1)</sup>
  - Optional Boot Code Section with Independent Lock Bits
    - In-System Programming by On-chip Boot Program
    - True Read-While-Write Operation
  - Programming Lock for Software Security
- Atmel® QTouch® Library Support
  - Capacitive Touch Buttons, Sliders and Wheels
  - QTouch and QMatrix® Acquisition
  - Up to 64 sense channels

- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
  - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Six PWM Channels
  - 8-channel 10-bit ADC in TQFP and QFN/MLF package
    - Temperature Measurement
  - 6-channel 10-bit ADC in PDIP Package
    - Temperature Measurement
  - Two Master/Slave SPI Serial Interface
  - One Programmable Serial USART
  - One Byte-oriented 2-wire Serial Interface (Philips I<sup>2</sup>C compatible)
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - One On-chip Analog Comparator
  - Interrupt and Wake-up on Pin Change
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated Oscillator
  - External and Internal Interrupt Sources
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
- I/O and Packages
  - 23 Programmable I/O Lines
  - 28-pin PDIP, 32-lead TQFP, 28-pad QFN/MLF and 32-pad QFN/MLF
- Operating Voltage:
  - 1.8 - 5.5V
- Temperature Range:
  - -40°C to 105°C
- Speed Grade:
  - 0 - 4MHz @ 1.8 - 5.5V
  - 0 - 10MHz @ 2.7 - 5.5V
  - 0 - 20MHz @ 4.5 - 5.5V
- Power Consumption at 1MHz, 1.8V, 25°C
  - Active Mode: 0.2mA
  - Power-down Mode: 0.1µA
  - Power-save Mode: 0.75µA (Including 32kHz RTC)

Features	ATmega328/P
Pin Count	28/32
Flash (Bytes)	32K
SRAM (Bytes)	2K
EEPROM (Bytes)	1K
Interrupt Vector Size (instruction word/vector)	1/1/2
General Purpose I/O Lines	23
SPI	2
TWI (I <sup>2</sup> C)	1
USART	1
ADC	10-bit 15kSPS
ADC Channels	8
8-bit Timer/Counters	2
16-bit Timer/Counters	1

and support a real Read-While-Write Self-Programming mechanism. There is a separate Boot Loader Section, and the SPM instruction can only execute from there. In , there is no Read-While-Write support and no separate Boot Loader Section. The SPM instruction can execute from the entire Flash.



Pinout diagram for a 28-pin DIP package, showing pin assignments for various pins. The diagram includes a legend for pin colors:

- Power (orange)
- Ground (black)
- Programming/debug (dark blue)
- Digital (light blue)
- Analog (green)
- Crystal/Osc (grey)

(PCINT14/RESET) PC6	1	28	PC5 (ADC5/SCL/PCINT13)
(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12)
(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11)
(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10)
(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9)
(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8)
VCC	7	22	GND
GND	8	21	AREF
(PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC
(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5)
(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4)
(PCINT22/OC0A/AIN0) PD6	12	17	PB3 (MOSI/OC2A/PCINT3)
(PCINT23/AIN1) PD7	13	16	PB2 (SS/OC1B/PCINT2)
(PCINT0/CLK0/ICP1) PB0	14	15	PB1 (OC1A/PCINT1)

## LAMPIRAN 5

### Datasheet DS18B20



[www.dalsemi.com](http://www.dalsemi.com)

#### FEATURES

- Unique 1-Wire interface requires only one port pin for communication
- Multidrop capability simplifies distributed temperature sensing applications
- Requires no external components
- Can be powered from data line. Power supply range is 3.0V to 5.5V
- Zero standby power required
- Measures temperatures from -55°C to +125°C. Fahrenheit equivalent is -67°F to +257°F
- ±0.5°C accuracy from -10°C to +85°C
- Thermometer resolution is programmable from 9 to 12 bits
- Converts 12-bit temperature to digital word in 750 ms (max.)
- User-definable, nonvolatile temperature alarm settings
- Alarm search command identifies and addresses devices whose temperature is outside of programmed limits (temperature alarm condition)
- Applications include thermostatic controls, industrial systems, consumer products, thermometers, or any thermally sensitive system

#### DESCRIPTION

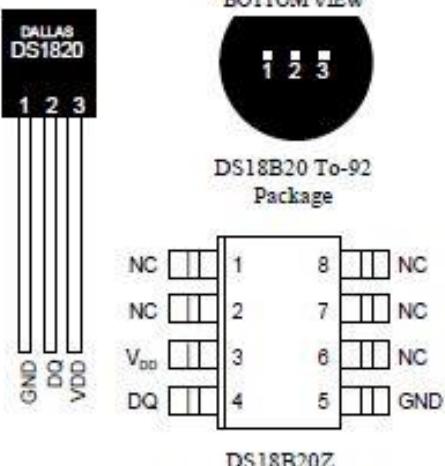
The DS18B20 Digital Thermometer provides 9 to 12-bit (configurable) temperature readings which indicate the temperature of the device.

Information is sent to/from the DS18B20 over a 1-Wire interface, so that only one wire (and ground) needs to be connected from a central microprocessor to a DS18B20. Power for reading, writing, and performing temperature conversions can be derived from the data line itself with no need for an external power source.

Because each DS18B20 contains a unique silicon serial number, multiple DS18B20s can exist on the same 1-Wire bus. This allows for placing temperature sensors in many different places. Applications where this feature is useful include HVAC environmental controls, sensing temperatures inside buildings, equipment or machinery, and process monitoring and control.

#### PRELIMINARY DS18B20 Programmable Resolution 1-Wire® Digital Thermometer

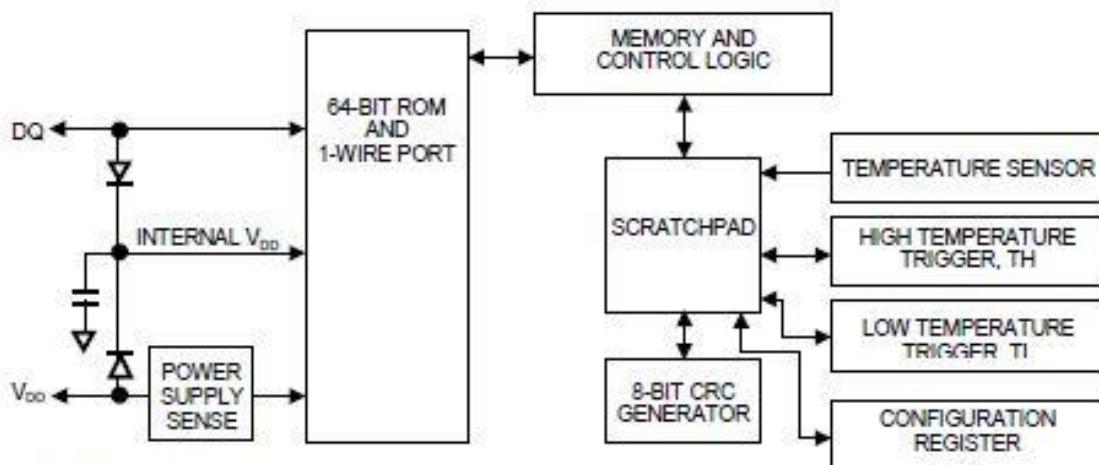
#### PIN ASSIGNMENT



#### PIN DESCRIPTION

GND	- Ground
DQ	- Data In/Out
VDD	- Power Supply Voltage
NC	- No Connect

DS18B20 BLOCK DIAGRAM Figure 1



### PARASITE POWER

The block diagram (Figure 1) shows the parasite-powered circuitry. This circuitry "steals" power whenever the DQ or V<sub>DD</sub> pins are high. DQ will provide sufficient power as long as the specified timing and voltage requirements are met (see the section titled "1-Wire Bus System"). The advantages of parasite power are twofold: 1) by parasitizing off this pin, no local power source is needed for remote sensing of temperature, and 2) the ROM may be read in absence of normal power.

In order for the DS18B20 to be able to perform accurate temperature conversions, sufficient power must be provided over the DQ line when a temperature conversion is taking place. Since the operating current of the DS18B20 is up to 1.5 mA, the DQ line will not have sufficient drive due to the 5k pullup resistor. This problem is particularly acute if several DS18B20s are on the same DQ and attempting to convert simultaneously.

There are two ways to assure that the DS18B20 has sufficient supply current during its active conversion cycle. The first is to provide a strong pullup on the DQ line whenever temperature conversions or copies to the E<sup>2</sup> memory are taking place. This may be accomplished by using a MOSFET to pull the DQ line directly to the power supply as shown in Figure 2. The DQ line must be switched over to the strong pullup within 10  $\mu$ s maximum after issuing any protocol that involves copying to the E<sup>2</sup> memory or initiates temperature conversions. When using the parasite power mode, the V<sub>DD</sub> pin must be tied to ground.

Another method of supplying current to the DS18B20 is through the use of an external power supply tied to the V<sub>DD</sub> pin, as shown in Figure 3. The advantage to this is that the strong pullup is not required on the DQ line, and the bus master need not be tied up holding that line high during temperature conversions. This allows other data traffic on the 1-Wire bus during the conversion time. In addition, any number of DS18B20s may be placed on the 1-Wire bus, and if they all use external power, they may all simultaneously perform temperature conversions by issuing the Skip ROM command and then issuing the Convert T command. Note that as long as the external power supply is active, the GND pin may not be floating.

The use of parasite power is not recommended above 100°C, since it may not be able to sustain communications given the higher leakage currents the DS18B20 exhibits at these temperatures. For applications in which such temperatures are likely, it is strongly recommended that V<sub>DD</sub> be applied to the DS18B20.

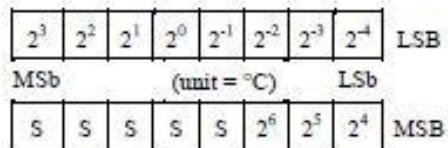
- - -

## OPERATION - MEASURING TEMPERATURE

The core functionality of the DS18B20 is its direct-to-digital temperature sensor. The resolution of the DS18B20 is configurable (9, 10, 11, or 12 bits), with 12-bit readings the factory default state. This equates to a temperature resolution of 0.5°C, 0.25°C, 0.125°C, or 0.0625°C. Following the issuance of the Convert T [44h] command, a temperature conversion is performed and the thermal data is stored in the scratchpad memory in a 16-bit, sign-extended two's complement format. The temperature information can be retrieved over the 1-Wire interface by issuing a Read Scratchpad [BEh] command once the conversion has been performed. The data is transferred over the 1-Wire bus, LSB first. The MSB of the temperature register contains the "sign" (S) bit, denoting whether the temperature is positive or negative.

Table 2 describes the exact relationship of output data to measured temperature. The table assumes 12-bit resolution. If the DS18B20 is configured for a lower resolution, insignificant bits will contain zeros. For Fahrenheit usage, a lookup table or conversion routine must be used.

Temperature/Data Relationships Table 2



TEMPERATURE	DIGITAL OUTPUT (Binary)	DIGITAL OUTPUT (Hex)
+125°C	0000 0111 1101 0000	07D0h
+85°C	0000 0101 0101 0000	0550h*
+25.0625°C	0000 0001 1001 0001	0191h
+10.125°C	0000 0000 1010 0010	00A2h
+0.5°C	0000 0000 0000 1000	0008h
0°C	0000 0000 0000 0000	0000h
-0.5°C	1111 1111 1111 1000	FFF8h
-10.125°C	1111 1111 0101 1110	FF5Eh
-25.0625°C	1111 1110 0110 1111	FF6Fh
-55°C	1111 1100 1001 0000	FC90h

\*The power on reset register value is +85°C.

## OPERATION - ALARM SIGNALING

After the DS18B20 has performed a temperature conversion, the temperature value is compared to the trigger values stored in TH and TL. Since these registers are 8-bit only, bits 9-12 are ignored for comparison. The most significant bit of TH or TL directly corresponds to the sign bit of the 16-bit temperature register. If the result of a temperature measurement is higher than TH or lower than TL, an alarm flag inside the device is set. This flag is updated with every temperature measurement. As long as the alarm flag is set, the DS18B20 will respond to the alarm search command. This allows many DS18B20s to be connected in parallel doing simultaneous temperature measurements. If somewhere the temperature exceeds the limits, the alarming device(s) can be identified and read immediately without having to read non-alarming devices.

## LAMPIRAN 6

### Datasheet *Real Time Clock (DS1302)*



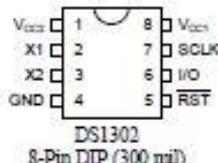
### DS1302 Trickle Charge Timekeeping Chip

[www.dalsemi.com](http://www.dalsemi.com)

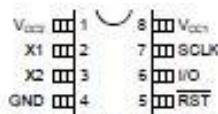
#### FEATURES

- Real time clock counts seconds, minutes, hours, date of the month, month, day of the week, and year with leap year compensation valid up to 2100
- 31 x 8 RAM for scratchpad data storage
- Serial I/O for minimum pin count
- 2.0–5.5V full operation
- Uses less than 300 nA at 2.0V
- Single-byte or multiple-byte (burst mode) data transfer for read or write of clock or RAM data
- 8-pin DIP or optional 8-pin SOICs for surface mount
- Simple 3-wire interface
- TTL-compatible ( $V_{CC} = 5V$ )
- Optional industrial temperature range  $-40^{\circ}C$  to  $+85^{\circ}C$
- DS1202 compatible
- Recognized by Underwriters Laboratory

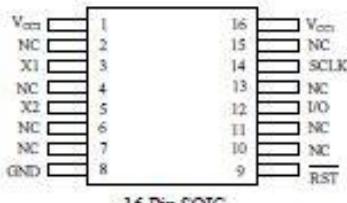
#### PIN ASSIGNMENT



DS1302  
8-Pin DIP (300 mil)



DS1302S 8-Pin SOIC (200 mil)  
DS1302Z 8-Pin SOIC (150 mil)



16-Pin SOIC

#### ORDERING INFORMATION

PART #	DESCRIPTION
DS1302	8-Pin DIP
DS1302N	8-Pin DIP (Industrial)
DS1302S	8-Pin SOIC (200 mil)
DS1302SN	8-Pin SOIC (Industrial)
DS1302Z	8-Pin SOIC (150 mil)
DS1302ZN	8-Pin SOIC (Industrial)
DS1302S-16	16-Pin SOIC (300 mil)
DS1302SN-16	16-Pin SOIC (Industrial)

#### PIN DESCRIPTION

X1, X2	– 32.768 kHz Crystal Pins
GND	– Ground
RST	– Reset
I/O	– Data Input/Output
SCLK	– Serial Clock
$V_{CC1}, V_{CC2}$	– Power Supply Pins

#### DESCRIPTION

The DS1302 Trickle Charge Timekeeping Chip contains a real time clock/calendar and 31 bytes of static RAM. It communicates with a microprocessor via a simple serial interface. The real time clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The end of the month date is automatically adjusted for months with less than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with an AM/PM indicator.

Interfacing the DS1302 with a microprocessor is simplified by using synchronous serial communication.

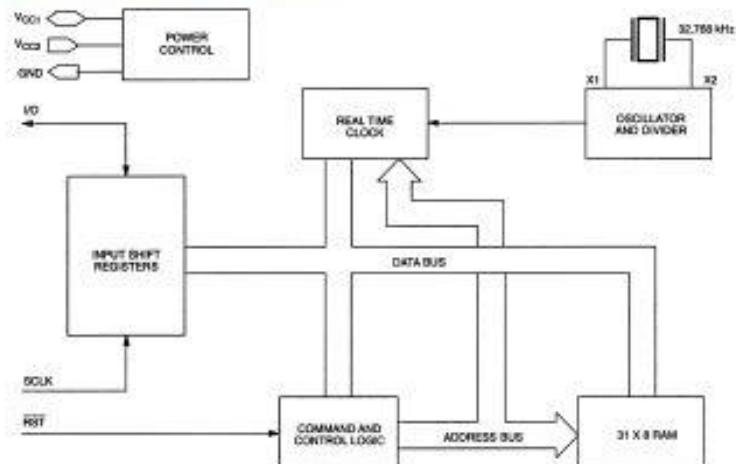
Only three wires are required to communicate with the clock/RAM: (1)  $\overline{\text{RST}}$  (Reset), (2) I/O (Data line), and (3) SCLK (Serial clock). Data can be transferred to and from the clock/RAM 1 byte at a time or in a burst of up to 31 bytes. The DS1302 is designed to operate on very low power and retain data and clock information on less than 1 microwatt.

The DS1302 is the successor to the DS1202. In addition to the basic timekeeping functions of the DS1202, the DS1302 has the additional features of dual power pins for primary and back-up power supplies, programmable trickle charger for  $V_{CC1}$ , and seven additional bytes of scratchpad memory.

## OPERATION

The main elements of the Serial Timekeeper are shown in Figure 1: shift register, control logic, oscillator, real time clock, and RAM.

**DS1302 BLOCK DIAGRAM Figure 1**



## SIGNAL DESCRIPTIONS

$V_{CC1}$  –  $V_{CC1}$  provides low power operation in single supply and battery operated systems as well as low power battery backup. In systems using the trickle charger, the rechargeable energy source is connected to this pin.

$V_{CC2}$  –  $V_{CC2}$  is the primary power supply pin in a dual supply configuration.  $V_{CC1}$  is connected to a backup source to maintain the time and date in the absence of primary power.

The DS1302 will operate from the larger of  $V_{CC1}$  or  $V_{CC2}$ . When  $V_{CC2}$  is greater than  $V_{CC1} + 0.2V$ ,  $V_{CC2}$  will power the DS1302. When  $V_{CC2}$  is less than  $V_{CC1}$ ,  $V_{CC1}$  will power the DS1302.

**SCLK (Serial Clock Input)** – SCLK is used to synchronize data movement on the serial interface.

**I/O (Data Input/Output)** – The I/O pin is the bi-directional data pin for the 3-wire interface.

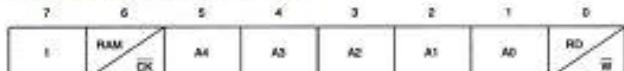
**$\overline{\text{RST}}$  (Reset)** – The reset signal must be asserted high during a read or a write.

**X1, X2** – Connections for a standard 32.768 kHz quartz crystal. The internal oscillator is designed for operation with a crystal having a specified load capacitance of 6 pF. For more information on crystal selection and crystal layout considerations, please consult Application Note 58, "Crystal Considerations with Dallas Real Time Clocks." The DS1302 can also be driven by an external 32.768 kHz oscillator. In this configuration, the X1 pin is connected to the external oscillator signal and the X2 pin is floated.

### COMMAND BYTE

The command byte is shown in Figure 2. Each data transfer is initiated by a command byte. The MSB (Bit 7) must be a logic 1. If it is 0, writes to the DS1302 will be disabled. Bit 6 specifies clock/calendar data if logic 0 or RAM data if logic 1. Bits 1 through 5 specify the designated registers to be input or output, and the LSB (bit 0) specifies a write operation (input) if logic 0 or read operation (output) if logic 1. The command byte is always input starting with the LSB (bit 0).

ADDRESS/COMMAND BYTE Figure 2



### RESET AND CLOCK CONTROL

All data transfers are initiated by driving the RST input high. The RST input serves two functions. First, RST turns on the control logic which allows access to the shift register for the address/command sequence. Second, the RST signal provides a method of terminating either single byte or multiple byte data transfer.

A clock cycle is a sequence of a falling edge followed by a rising edge. For data inputs, data must be valid during the rising edge of the clock and data bits are output on the falling edge of clock. If the RST input is low all data transfer terminates and the I/O pin goes to a high impedance state. Data transfer is illustrated in Figure 3. At power-up, RST must be a logic 0 until  $V_{CC} > 2.0$  volts. Also SCLK must be at a logic 0 when RST is driven to a logic 1 state.

### DATA INPUT

Following the eight SCLK cycles that input a write command byte, a data byte is input on the rising edge of the next eight SCLK cycles. Additional SCLK cycles are ignored should they inadvertently occur. Data is input starting with bit 0.

### DATA OUTPUT

Following the eight SCLK cycles that input a read command byte, a data byte is output on the falling edge of the next eight SCLK cycles. Note that the first data bit to be transmitted occurs on the first falling edge after the last bit of the command byte is written. Additional SCLK cycles retransmit the data bytes should they inadvertently occur so long as RST remains high. This operation permits continuous burst mode read capability. Also, the I/O pin is tri-stated upon each rising edge of SCLK. Data is output starting with bit 0.

### BURST MODE

Burst mode may be specified for either the clock/calendar or the RAM registers by addressing location 31 decimal (address/command bits 1 through 5 = logic 1). As before, bit 6 specifies clock or RAM and bit 0