Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR – EE 184801

# SISTEM KONTROL SUHU PADA PROSES FERMENTASI RUSIP MENGGUNAKAN METODE LOGIKA FUZZY BERDASARKAN TINGKAT KEASAMAN


Axel William
NRP 07111640000043

Dosen Pembimbing
Dr. Muhammad Rivai, S.T., M.T.
Fajar Budiman, S.T., M.Sc.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro Dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2020

*Halaman ini sengaja dikosongkan*

**TUGAS AKHIR – EE 184801**

# SISTEM KONTROL SUHU PADA PROSES FERMENTASI RUSIP MENGGUNAKAN METODE LOGIKA FUZZY BERDASARKAN TINGKAT KEASAMAN

Axel William
NRP. 07111640000043

Dosen Pembimbing
Dr. Muhammad Rivai, S.T, M.T.
Fajar Budiman, S.T., M.Sc.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro Dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2020

*Halaman ini sengaja dikosongkan*

**FINAL PROJECT – EE 184801**

# TEMPERATURE CONTROL SYSTEM IN RUSIP FERMENTATION PROCESS USING FUZZY LOGIC METHOD BASED ON ACIDITY LEVEL

Axel William
NRP 07111640000043

Supervisor
Dr. Muhammad Rivai, S.T., M.T.
Fajar Budiman, S.T., M.Sc.

*DEPARTMENT OF ELECTRICAL ENGINEERING*
*Faculty of Intelligent Electrical and Informatics Technology*
Institut Teknologi Sepuluh Nopember
Surabaya 2020

*Halaman ini sengaja dikosongkan*

# PERNYATAAN KEASLIAN
# TUGAS AKHIR

Dengan ini menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul "**Sistem Kontrol Suhu Pada Proses Fermentasi Rusip Menggunakan Metode Logika Fuzzy Berdasarkan Tingkat Keasaman**" adalah hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Juli 2020

Axel William
NRP. 07111640000043

*Halaman ini sengaja dikosongkan*

**SISTEM KONTROL SUHU PADA PROSES FERMENTASI RUSIP MENGGUNAKAN METODE LOGIKA FUZZY BERDASARKAN TINGKAT KEASAMAN**

# TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan Untuk
Memperoleh Gelar Sarjana Teknik Elektro

Pada

Bidang Studi Elektronika
Departemen Teknik Elektro
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember

Menyetujui:

Dosen Pembimbing I

Dr. Muhammad Rivai, S.T., M.T.
NIP. 196904261994031003

**SURABAYA,**
**JUNI 2020**

*Halaman ini sengaja dikosongkan*

**SISTEM KONTROL SUHU PADA PROSES FERMENTASI RUSIP MENGGUNAKAN METODE LOGIKA FUZZY BERDASARKAN TINGKAT KEASAMAN**

# TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan Untuk
Memperoleh Gelar Sarjana Teknik Elektro

Pada

Bidang Studi Elektronika
Departemen Teknik Elektro
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember

Menyetujui:

Dosen Pembimbing II

Fajar Budiman, S.T., M.Sc.
NIP. 198607072014041001

**SURABAYA,**
**JUNI 2020**

*Halaman ini sengaja dikosongkan*

**SISTEM KONTROL SUHU PADA PROSES FERMENTASI RUSIP MENGGUNAKAN METODE LOGIKA FUZZY BERDASARKAN TINGKAT KEASAMAN**

Axel William
07111640000043

Dosen Pembimbing : 1. Dr. Muhammad Rivai, S.T., M.T.
2. Fajar Budiman, S.T., M.Sc.

# ABSTRAK

Rusip merupakan makanan khas asal kepulauan Bangka Belitung berbahan dasar ikan teri segar, garam, dan gula aren yang difermentasikan. Melalui proses fermentasi, Rusip dapat bertahan lebih lama karena bakteri asam laktat yang memecah senyawa glukosa menjadi asam laktat. Semakin banyak asam laktat yang dihasilkan akan menyebabkan Rusip menjadi bersifat asam, dan bakteri patogen tidak dapat mengontaminasi Rusip. Namun, metode pengelolaan Rusip dalam berbagai UMKM selama ini masih bersifat konvensional dimana proses fermentasi berlangsung cukup lama, memakan waktu dua minggu. Karena itu diperlukanlah sebuah sistem kontrol proses fermentasi yang dapat mempercepat perkembangbiakan bakteri asam laktat dengan tetap menjaga keseragaman kualitas akhir produk Rusip. Pada penelitian ini akan dibuat sistem kontrol suhu dengan menggunakan dua buah sensor diantaranya sensor suhu dan pH. Jenis sensor suhu yang digunakan adalah thermistor DS18B20 yang telah dilengkapi lapisan tahan air, dimana dapat mengukur suhu -55℃ – 125℃ dengan akurasi maksimum ±2,0℃. Sedangkan jenis sensor pH yang digunakan adalah elektroda probe celup dengan spesifikasi pengukuran tingkat keasaman 0 – 14 dan rentang operasi 0 – 80℃. Dengan memanfaatkan sensor suhu dan pH ditambah penerapan metode pengontrolan logika fuzzy, sistem kontrol dapat mengoptimalkan besarnya nilai temperatur terbaru berdasarkan pendekatan linguistik manusia terhadap pembacaan parameter kontrol baik itu tingkat keasaman dan suhu sebelumnya. Secara keseluruhan sistem berhasil menjaga kualitas Rusip serta mempercepat proses fermentasi 28,57% lebih cepat daripada proses fermentasi spontan.

**Kata Kunci :** *Bakteri Asam Laktat, Fermentasi Rusip, Logika Fuzzy, Sensor PH.*

*Halaman ini sengaja dikosongkan*

***TEMPERATURE CONTROL SYSTEM IN RUSIP FERMENTATION PROCESS USING FUZZY LOGIC METHOD BASED ON ACIDITY LEVEL***

Axel William
07111640000043

*Supervisor* : 1. Dr. Muhammad Rivai, S.T., M.T.
2. Fajar Budiman, S.T., M.Sc.

## *ABSTRACT*

Rusip is a typical food from the islands of Bangka Belitung made from fermented fresh anchovy, salt, and palm sugar. Through the fermentation process, Rusip can last longer because of lactic acid bacteria that break down glucose compounds into lactic acid. The more lactic acid produced will cause the Rusip to be acidic, and pathogenic bacteria cannot contaminate the Rusip. However, the management method of Rusip in various Micro Small and Medium Enterprises is still using a conventional method, where the fermentation process takes quite a long time, taking two weeks. Therefore we need a fermentation process control system that can accelerate the proliferation of lactic acid bacteria while maintaining uniformity in the final quality of Rusip products. In this research, a temperature control system will be using two types of sensors, including temperature and pH sensors. The type of temperature sensor used is the DS18B20 thermistor that has equipped with a waterproof layer, which can measure temperatures -55 ℃ to 125 ℃ with a maximum accuracy of ± 2,0 ℃. While the type of pH sensor used is a dip probe electrode with acidity measurement specifications 0-14 pH ranging and operating range 0 - 80 ℃, By utilizing temperature and pH sensors plus the application of a fuzzy logic control method, the control system can optimize the magnitude of the latest temperature values based on the individual linguistic approach to the reading of control parameters both acidity and temperature levels before. Overall, the system managed to maintain the quality of Rusip and accelerate the fermentation process by 28.57% faster than the spontaneous fermentation process.

***Keyword :*** *Fuzzy Logic, Lactic Acid Bacteria, PH Sensor, Rusip Fermentation.*

*Halaman ini sengaja dikosongkan*

# KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan YME karena atas rahmat dan penyertaan-Nya, penulis dapat mengerjakan dan menuntaskan laporan tugas akhir ini dengan judul "**SISTEM KONTROL SUHU PADA PROSES FERMENTASI RUSIP MENGGUNAKAN METODE LOGIKA FUZZY BERDASARKAN TINGKAT KEASAMAN**".

Tujuan dari pembuatan laporan tugas akhir ini tidak lain sebagai salah satu prasyarat kelulusan penyelesaian program pendidikan sarjana dan mata kuliah yang wajib ditempuh di Departemen Teknik Elektro Institut Teknologi Sepuluh Nopember.

Selama proses pengerjaan penulis menyadari sebagai manusia biasa tanpa bantuan, dukungan dan arahan dari banyak pihak tanpanya mungkin saja laporan tugas akhir ini tidak bisa diselesaikan terlebih ditengah pandemi COVID-19, oleh karenanya secara khusus penulis ingin mengucapkan terima kasih kepada:

1. Pak Dr. Muhammad Rivai, S.T., M.T. dan Pak Fajar Budiman, S.T., M.T. selaku dosen pembimbing yang telah banyak memberikan bantuan baik gagasan topik tugas akhir, bimbingan serta izin penggunaan laboratorium B402 kepada penulis selama mengerjakan tugas akhir.
2. Dosen penguji sidang tugas akhir bapak Ronny Mardiyanto, S.T., M.T., Ph.D., Ir. Tasripan, M.T., Muhammad Attamimi, B.Eng., M.Eng., Ph.D., dan Ir. Harris Pirngadi, M.T., yang telah memberi saran dan masukan terhadap tugas akhir ini.
3. Bapak Dedet Chandra Riawan, S.T., M.Eng, Ph.D. selaku Kepala Departemen Teknik Elektro ITS.
4. Seluruh dosen dan staf pegawai Departemen Teknik Elektro ITS.
5. Keluarga besar, orang tua, Papa dan Mama yang telah membesarkan penulis serta dukungan moral, materiil, dan ilmu yang diterapkan dalam pengerjaan tugas akhir.
6. Teman-teman elektronika sesama penghuni laboratium mikroelektronika dan sistem tertanam B202 dan teman seperjuangan tugas akhir laboratorium elektronika industri B403

7. Secara khusus Vincent dan Adrian teman satu *unit* apartemen yang telah membantu penulis dalam pembuatan alat tugas akhir.

Terlepas dari itu semua, penulis menyadari masih banyak kekurangan baik dari segi penyajian laporan dan materi oleh karenanya penulis menerima kritik dan saran yang diberikan demi pengembangan alat yang lebih baik. Akhir kata, semoga dengan diadakannya penelitian tugas akhir ini dapat memberikan manfaat bagi banyak pihak.

Surabaya, 27 Juli 2020

Axel William

# DAFTAR ISI

PERNYATAAN KEASLIAN TUGAS AKHIR ..... VII
ABSTRAK ..... XIII
*ABSTRACT* ..... XV
KATA PENGANTAR ..... XVII
DAFTAR ISI ..... XIX
DAFTAR GAMBAR ..... XXIII
DAFTAR TABEL ..... XXVII
BAB 1 PENDAHULUAN ..... 1
1.1 Latar Belakang ..... 1
1.2 Perumusan Masalah ..... 2
1.3 Batasan Masalah ..... 3
1.4 Tujuan Penelitian ..... 3
1.5 Metode Penelitian ..... 3
1.5.1 Studi Literasi Fermentasi Rusip ..... 3
1.5.2 Perancangan Perangkat Lunak Kontrol Logika Fuzzy ..... 4
1.5.3 Perancangan Sistem Sensor PH dan Suhu ..... 4
1.5.4 Perancangan Sistem Aktuator Heater ..... 4
1.5.5 Pengujian Keseluruhan Sistem Kontrol ..... 5
1.5.6 Analisa dan Evaluasi Sistem Kontrol ..... 5
1.5.7 Penyusunan Buku Tugas Akhir ..... 5
1.5.8 Penyusunan Jurnal Pomits ..... 5
1.6 Sistematika Penulisan ..... 6
1.7 Relevansi ..... 6
BAB 2 TINJAUAN PUSTAKA ..... 9
2.1 Fermentasi ..... 9
2.2 Rusip ..... 11
2.3 pH (*Puissance d'Hydrogen*) ..... 13
2.4 PH Meter ..... 15
2.5 Logika Fuzzy ..... 17
2.6 Sensor Temperatur DS18B20 ..... 20
2.7 Liquid Crystal Display Thin Film Transistor (LCD TFT) ..... 22
2.8 Arduino Uno ..... 23
2.8.1 Sumber Daya ..... 24
2.8.2 Input dan Output ..... 25
2.8.3 Pin Komunikasi ..... 26
2.8.4 Pemrograman ..... 26
2.9 Arduino Mega 2560 ..... 27

2.10 RTC DS3231 ..... 29
2.11 Modul Data Logger Kartu Mikro SD ..... 30
2.12 IRF540N ..... 32
2.13 2N3904 ..... 32
2.14 Kawat Nikrom ..... 33
BAB 3 PERANCANGAN SISTEM ..... 37
3.1 Diagram Blok Sistem ..... 38
3.2 Perancangan Perangkat Keras ..... 40
3.2.1 Arduino Uno dengan TFT LCD ..... 40
3.2.1 Antarmuka Arduino Uno dengan Arduino Mega ..... 42
3.2.2 Pembacaan Sensor DS18B20 ..... 42
3.2.3 Pembacaan Sensor PH ..... 43
3.2.4 Arduino Mega dengan Modul RTC DS3231 ..... 44
3.2.5 Arduino Mega dengan Modul SD Card ..... 45
3.2.6 Arduino Uno dengan Aktuator Pemanas ..... 45
3.2.7 Arduino Uno ..... 47
3.2.8 Pemanas Kawat Nikrom ..... 47
3.3 Perancangan Perangkat Lunak ..... 47
3.3.1 Pembacaan Sensor Suhu DS18B20 ..... 49
3.3.2 Pembacaan Sensor PH ..... 50
3.3.3 Penyusunan Lookup Table Konversi PH Menjadi Setpoint Suhu ..... 51
3.3.4 Pengiriman Data antara Arduino Mega dengan Arduino Uno ..... 53
3.3.5 Tampilan Interface Parameter Kontrol Pada LCD TFT . 56
3.3.6 Sistem Kontrol Logika Fuzzy ..... 58
BAB 4 PENGUJIAN DAN ANALISIS ..... 65
4.1 Kalibrasi Sistem Sensor ..... 67
4.1.1 Kalibrasi Sensor Suhu dengan Termometer ..... 68
4.1.2 Kalibrasi Sensor PH dengan Cairan PH Buffer ..... 70
4.2 Pengujian Sistem Aktuator Pemanas ..... 72
4.3 Pengujian *Lookup Table* Konversi PH Menjadi Setpoint Suhu ..... 75
4.4 Pengujian Sistem Kontrol Logika Fuzzy ..... 75
4.5 Pengujian Sistem Kontrol Keseluruhan ..... 80
4.6 Analisa Keseluruhan Sistem ..... 89
BAB 5 PENUTUP ..... 93
5.1 Kesimpulan ..... 93
5.2 Saran ..... 94
DAFTAR PUSTAKA ..... 95

LAMPIRAN A ..... 99
LAMPIRAN B ..... 101
LAMPIRAN C ..... 111
LAMPIRAN D ..... 119
LAMPIRAN E ..... 123
LAMPIRAN F ..... 127
LAMPIRAN G ..... 133
LAMPIRAN H ..... 139
LAMPIRAN I ..... 148
LAMPIRAN J ..... 150
LAMPIRAN K ..... 154
LAMPIRAN L ..... 159
LAMPIRAN M ..... 166
BIODATA PENULIS ..... 179

*Halaman ini sengaja dikosongkan*

# DAFTAR GAMBAR

Gambar 2. 1. Diagram Pemecahan Molekul Glukosa Menjadi Molekul Asam Laktat Pada Proses Fermentasi Bakteri Asam Laktat ..... 10
Gambar 2. 2. Grafik Pengaruh Pengingkatan Temperatur dan pH terhadap Pertumbuhan Bakteri Asam Laktat ..... 12
Gambar 2. 3. Rusip Dalam Kemasan ..... 12
Gambar 2. 5. Skala pH ..... 15
Gambar 2. 4. Proses Pembuatan Fermentasi Rusip ..... 13
Gambar 2. 6. Skematik Representasi pH Meter Terdiri Dari Elektroda Kaca Terkopel Dengan Elektroda Referensi Terpisah (a) Dan Sistem Gabungan (b) ..... 16
Gambar 2. 7. Arsitektur Logika Fuzzy ..... 17
Gambar 2. 8. Fungsi Keanggotaan Logika Fuzzy Segitiga, Trapesium dan Gaussian ..... 18
Gambar 2. 9. Proses Inferensi Fuzzy Tipe Madani ..... 20
Gambar 2. 10. Metode Defuzzifikasi Madani ..... 20
Gambar 2. 11. Konfigurasi Sensor Temperatur DS18B20 ..... 21
Gambar 2. 12. Lapisan Penyusun LCD TFT ..... 22
Gambar 2. 13. Arduino Uno ..... 23
Gambar 2. 14. Diagram Pin Arduino Uno ..... 26
Gambar 2. 15. Arduino Mega 2560 ..... 27
Gambar 2. 16. Pin Diagram RTC DS3231 ..... 29
Gambar 2. 17. Pin Diagram Modul Kartu Mikro SD ..... 31
Gambar 2. 18. Pin Diagram IRF540N ..... 34
Gambar 2. 19. Diagram Pin Arduino Mega ..... 35
Gambar 2. 20. Pin Diagram 2N3904 ..... 36
Gambar 2. 21. Kawat Nikrom ..... 36
Gambar 3. 1. Skema Sistem Keseluruhan ..... 37
Gambar 3. 2. Diagram Blok Sistem Keseluruhan ..... 38
Gambar 3. 3. Antarmuka Arduino Uno dengan LCD TFT ..... 40
Gambar 3. 4. Skema Rangkaian Sistem Interface antara Arduino Uno dengan LCD TFT ..... 41
Gambar 3. 5. Antarmuka Arduino Mega dengan Arduino Uno ..... 42
Gambar 3. 6. Antarmuka Arduino Uno dengan Sensor Suhu ..... 43
Gambar 3. 7. Antarmuka Arduino Mega dengan Sensor PH ..... 44
Gambar 3. 8. Antarmuka Arduino Mega dengan Modul RTC DS3231 ..... 44

Gambar 3. 9. Antarmuka Arduino Mega dengan Modul Datalogging SD Card ..... 45
Gambar 3. 10. Antarmuka Arduino Mega dengan Aktuator Pemanas 46
Gambar 3. 11. *Flowchart* Sederhana Sistem Kontrol Keseluruhan .... 48
Gambar 3. 12. Program Pembacaan Sensor Suhu DS18B20 ..... 49
Gambar 3. 13. Program Pembacaan Sensor PH ..... 51
Gambar 3. 14. *Flowchart* Lookup Table Mengonversikan Parameter PH Menjadi Parameter Suhu Setpoint ..... 52
Gambar 3. 15. Program Pengiriman Data Arduino Mega Subsistem Kontrol..... 53
Gambar 3. 16. Program Penerima Data Kontrol Arduino Uno Subsistem Interface..... 54
Gambar 3. 17. Program Pembaca Penanda Awal dan Akhir Data ..... 55
Gambar 3. 18. Program Pemisah Data Parameter Kontrol..... 55
Gambar 3. 19. Tampilan Menu Utama Antarmuka..... 56
Gambar 3. 20. Tampilan Antarmuka Submenu Temperatur ..... 57
Gambar 3. 21. Tampilan Antarmuka Submenu PH..... 57
Gambar 3. 22. Tampilan Antarmuka Submenu Grafik Historis..... 58
Gambar 3. 23. *Flowchart* Sistem Logika Fuzzy ..... 60
Gambar 3. 24. Fungsi Keanggotaan Error..... 60
Gambar 3. 25. Fungsi Keanggotaan Δerror..... 61
Gambar 3. 26. Fungsi Keanggotaan Output Nilai PWM Pemanas ..... 63
Gambar 4. 1. Realisasi Subsistem Kontrol Utama (Dalam)..... 65
Gambar 4. 2. Realisasi Subsistem Interface (Dalam)..... 66
Gambar 4. 3. Realisasi Box Sistem Kontrol (Luar) ..... 66
Gambar 4. 4. Realisasi Pemanas Kawat Nikrom Bagian Dalam (a) dan Realisasi Pemanas Kawat Nikrom Bagian Luar (b). 67
Gambar 4. 5. Realisasi Bioreaktor Fermentasi Rusip ..... 67
Gambar 4. 6. Instrumen Termometer Digital..... 68
Gambar 4. 7. Grafik Perbandingan Suhu Termometer dengan Sensor Suhu..... 70
Gambar 4. 8. Cairan PH Buffer Solution ..... 70
Gambar 4. 9. Hasil Persamaan Linear Kalibrasi Sensor pH..... 71
Gambar 4. 10. Grafik Perbandingan Pengukuran Sensor PH dengan Cairan PH Buffer..... 72
Gambar 4. 11. Hubungan Antara Output Nilai PWM Terhadap Suhu yang Dihasilkan ..... 74
Gambar 4. 12. Hubungan Antara Nilai PWM Terhadap Keluaran Tegangan IRF540N ..... 74

Gambar 4. 13. Grafik Respon Sistem Kontrol Logika Fuzzy Pada Setpoint Suhu 30℃ dengan Suhu Awal 28℃ ..... 76
Gambar 4. 14. Grafik Respon Sistem Kontrol Logika Fuzzy pada Setpoint Suhu 34℃ dengan Suhu Awal 30,5℃ ..... 76
Gambar 4. 15. Grafik Respon Sistem Kontrol Logika Fuzzy Pada Setpoint Suhu 37℃ dengan Suhu Awal 30℃ ..... 80
Gambar 4. 16. Realisasi Keseluruhan Sistem ..... 81
Gambar 4. 17. Ikan Teri Bahan Baku Utama Rusip ..... 83
Gambar 4. 18. Irisan Gula Merah pada Bagian (a) dan Garam Meja pada Bagian (b) ..... 83
Gambar 4. 19. Grafik Perubahan Harian Tingkat Keasaman Selama Proses Fermentasi Rusip ..... 86
Gambar 4. 20. Grafik Penurunan Tingkat Keasaman dengan Fermentasi Rusip Secara Spontan dan Tidak Spontan. 87
Gambar 4. 21. Grafik Perubahan Harian Suhu Selama Proses Fermenasi Rusip ..... 87
Gambar 4. 22. Hasil Akhir Rusip pada Hari ke 10 ..... 88
Gambar 4. 23. Kuisioner *Human Sensory Test* ..... 88
Gambar 4. 24. Data Keseluruhan Penilaian *Human Sensory Test* ..... 89

*Halaman ini sengaja dikosongkan*

# DAFTAR TABEL


Tabel 2. 1. Karakteristik Kandungan Rusip ..... 14
Tabel 2. 2. Spesifikasi Elektroda Kaca..... 16
Tabel 2. 3. Spesifikasi Modul pH Meter ..... 17
Tabel 2. 4. Spesifikasi Sensor Temperatur DS18B20 ..... 21
Tabel 2. 5. Spesifikasi Arduino Uno ..... 24
Tabel 2. 6. Spesifikasi Arduino Mega 2560 ..... 28
Tabel 2. 7. Spesifikasi RTC DS3231..... 30
Tabel 2. 8. Deskripsi Pin RTC DS3231 ..... 30
Tabel 2. 9. Spesifikasi Modul Kartu Micro SD..... 31
Tabel 3. 1. *Lookup Table* Konversi PH Menjadi Setpoint Suhu..... 52
Tabel 3. 2. Rule Base Pengaturan PWM Pemanas ..... 62
Tabel 4. 1. Hasil Kalibrasi Sensor Suhu Terhadap Termometer ..... 69
Tabel 4. 2. Hasil Pengukuran Sensor PH Terhadap Cairan PH Buffer 72
Tabel 4. 3. Pengaruh Nilai PWM Pemanas Terhadap Suhu..... 73
Tabel 4. 4. Hasil Konversi *Lookup Table*..... 75
Tabel 4. 5. Pengujian Kontrol Logika Fuzzy Pada Setpoint Suhu 30℃ dengan Suhu Awal 28℃ ..... 77
Tabel 4. 6. Pengujian Kontrol Logika Fuzzy Pada Setpoint Suhu 34℃ dengan Suhu Awal 30,6℃ ..... 78
Tabel 4. 7. Pengujian Kontrol Logika Fuzzy pada Setpoint Suhu 37℃ dengan Suhu Awal 30℃ ..... 79
Tabel 4. 8. Rata-Rata Data Pengukuran Fermentasi Hari Kedua ..... 82
Tabel 4. 9. Rata-Rata Data Pengukuran Fermentasi Hari Kelima ..... 84
Tabel 4. 10. Rata-Rata Data Pengukuran Fermentasi Hari Kedelapan.. 85

*Halaman ini sengaja dikosongkan*

# BAB 1
# PENDAHULUAN

Tugas akhir merupakan suatu penelitian yang dilakukan sebagai persyaratan akademik yang wajb bagi setiap mahasiswa untuk menyelesaikan masa pendidikan bagi tingkatan sarjana di departemen teknik elektro Institut Teknologi Sepuluh Nopember (ITS). Topik yang akan dibahas pada tugas akhir ini mengenai pemanfaatan sistem kontrol suhu pada proses fermentasi rusip menggunakan metode logika fuzzy berdasarkan tingkat keasaman.

Pada bab ini akan menjelaskan hal-hal mendasar dari topik pelaksanaan tugas akhir. Hal-hal tersebut diataranya adalah latar belakang, perumusan masalah, tujuan penulisan, batasan masalah, metodologi penelitian, sistematika penulisan, dan relevansi penelitian dengan penelitian serupa dikemudian hari.

## 1.1 Latar Belakang

Sejak zaman pra sejarah manusia telah memanfaatkan proses fermentasi untuk memperpanjang jangka waktu konsumsi berbagai macam jenis makanan. Fermentasi tidak hanya mempertahankan makanan tetapi juga meningkatkan kualitas sensorik dari produk hasil akhir [1]. Kualitas sensorik sendiri melibatkan evaluasi kualitas organoleptik atau evaluasi sensorik yang melibatkan rasa, bau, penampilan, dan tekstur, diantara atribut lainnya [2]. Sudah cukup banyak metode fermentasi tradisional berbasis bakteri asam laktat, terutama pada fermentasi ikan di berbagai negara Asia, tidak terkecuali Indonesia seperti Rusip. Rusip merupakan makanan tradisional khas dari daerah Bangka-Belitung berupa pengawetan ikan laut yang berukuran kecil terutama berbahan baku ikan teri yang diolah dengan cara fermentasi dengan penambahan garam dan gula aren dalam jumlah tertentu [3]. Beraneka macam cara pembuatan rusip, di antaranya ada yang menggunakan gula aren padat, dan gula aren cair [4].

Seiring dengan berjalannya waktu tingkat pertumbuhan industri perikanan bergerak secara linear dimana fakta tersebut akan menguntungkan bagi Indonesia selaku negara maritim. Berdasarkan data produksi perikanan nasional yang dihimpun oleh Badan Pusat

Statistik, sejak tahun 2011 hingga 2016 produksi perikanan nasional mengalami pertumbuhan sebesar 170,5%. Tepatnya terjadi pertumbuhan dari 13,64 juta ton metrik menjadi 23,51 juta ton metrik. Dengan meningkatnya produksi industri perikanan akan mengakibatkan ikutnya peningkatan limbah ikan termasuk limbah ikan teri yang dapat dimanfaatkan sebagai bahan baku rusip. Namun, selama ini di dalam proses pembuatan rusip belum dilengkapi dengan sistem kontrol sehingga mengakibatkan kualitas rusip yang dihasilkan tidak memiliki standarisasi hasil akhir yang sama serta waktu yang dibutuhkan untuk proses fermentasi masih bervariasi, dan memakan waktu cukup lama. Maka dari itu, diperlukanlah sistem kontrol yang efektif dan efisien untuk menjaga parameter lingkungan wadah fermentasi atau bioreaktor, dengan menggunakan dua buah sensor yakni sensor suhu dan pH.

Dikarenakan hampir seluruh proses fermentasi selalu melibatkan parameter suhu, maka parameter suhu dianggap sebagai parameter utama yang berperan langsung dalam keberlangsungan proses fermentasi. Kendati demikian parameter suhu tidak hanya mempengaruhi waktu fermentasi, tetapi juga mempengaruhi laju pertumbuhan [5]. Sedangkan sensor pH digunakan untuk menentukan kesesuaian kualitas hasil akhir rusip yang umumnya memiliki rasa yang asam atau berkadar pH rendah.

Melalui penerapan sistem kontrol ini, akan mempermudah pemilik atau pelaku usaha fermentasi rusip untuk meningkatkan produktivitas dengan mempercepat laju pertumbuhan dan perkembangbiakan bakteri asam laktat tanpa berkurangnya kualitas produk hasil akhir rusip tersebut, akibat keberhasilan dalam meminimalisir proses pembusukan oleh bakteri patogen yang mengkontaminasi rusip.

## 1.2 Perumusan Masalah

Perumusan masalah pada tugas akhir ini adalah sebagai berikut:

1. Bagaimana cara mempercepat proses fermentasi rusip dengan tetap menjaga kualitas akhir produk?
2. Bagaimana cara mengonversi parameter kadar pH menjadi parameter suhu sebagai set point kontrol?

3. Bagaimana pengendalian aktuator heater yang dapat menyesuaikan kebutuhan suhu fermentor dan mencegah terjadinya kerusakan akibat *overheating*?

## 1.3 Batasan Masalah

Perlu diketahui bahwasanya batasan dalam pelaksanaan tugas akhir ini, antara lain sebagai berikut:

1. Bahan baku rusip yang digunakan adalah ikan teri atau ikan bilis.
2. Aktuator yang digunakan hanya kawat nikelin dengan cara pemanasan bioreaktor secara radiasi melalui celah udara.
3. Pengujian dilakukan di dalam ruangan tertutup dilengkapi dengan pendingin udara.

## 1.4 Tujuan Penelitian

Adapun pelaksanaan penelitian tugas akhir ini bertujuan untuk:

1. Mengimplementasi metode logika fuzzy dalam sistem kontrol fermentasi rusip.
2. Penggunaan *lookup table* dan logika fuzzy pengkonversi kadar pH menjadi suhu sebagai set point kontrol.
3. Sistem kontrol fermentasi dapat mengendalikan aktuator heater yang berupa sinyal *Pulse Width Modulation* (PWM) sehingga dapat menyesuaikan kebutuhan suhu fermentor dan mencegah terjadinya kerusakan akibat overheating.

## 1.5 Metode Penelitian

Penelitian ini akan dilaksanakan dengan melalui beberapa tahap proses yang telah dirancang sebagai berikut :

### 1.5.1 Studi Literasi Fermentasi Rusip

Studi Literasi merupakan tahapan untuk menggali lebih dalam tentang aplikasi teknologi yang telah digunakan dan proses-proses yang terjadi dalam fermentasi rusip. Studi fermentasi rusip lebih ditekankan pada mengumpulkan data karakteristik rusip, proses pengolahan rusip, teknologi kontrol fermentasi dengan metode logika fuzzy, sensor pH sebagai sensor utama, dan sensor suhu sebagai sensor pendukung serta cara pembuatan bioreaktor pengolah rusip. Demi menjaga keabsahan tugas akhir ini dilakukan penyeleksian terhadap sumber informasi penunjang. sumber-sumber yang dikutip haruslah berasal dari berbagai paper, jurnal, buku, maupun artikel

insitutsi akademik terpecaya dengan rentang umur 5-10 tahun, dihitung dari tahun terbitan.

### 1.5.2 Perancangan Perangkat Lunak Kontrol Logika Fuzzy

Pada tahapan ini akan dilakukan perancangan dan pengembangan program kontrol logika fuzzy yang sesuai untuk diterapkan dalam proses fermentasi rusip yang ditinjau dari apa yang telah diperoleh dari tahapan studi literasi fermentasi rusip. Pembuatan program kontrol logika fuzzy akan terlebih dahulu disimulasikan dan diatur dalam aplikasi Matlab kemudian dilanjutkan dengan pengubahan parameter-parameter hasil simulasi menjadi bahasa pemrograman mikrokontroller arduino sehingga sistem kontrol yang dirancang memiliki performa yang *robust*, karakteristik *error steady state* yang kecil serta memudahkan implementasi ke perangkat keras.

### 1.5.3 Perancangan Sistem Sensor PH dan Suhu

Setelah dilakukan pemrograman kontroler logika fuzzy, tentunya dalam sebuah sistem kontrol tidak lengkap apabila hanya terdiri dari bagian kontroler saja dalam mengontrol proses fermentasi, diperlukanlah sistem *feedback element* atau elemen umpan balik yang terdiri dari sensor pH dan suhu. Sensor pH berfungsi untuk mengukur perubahan kadar pH dalam bioreaktor selama proses fermentasi berlangsung guna mengatur besaran titik acuan suhu yang akan diproses oleh sistem kontrol dimana suhu yang akan dihasilkan oleh sistem merupakan suhu optimal bagi pertumbuhan bakteri asam laktat, sedangkan sensor suhu bertugas untuk memberikan umpan balik besaran parameter suhu di dalam bioreaktor tetap terjaga dan sesuai antara nilai *setpoint* terhadap besaran keluaran yang dikeluarkan.

### 1.5.4 Perancangan Sistem Aktuator Heater

Agar sistem kontrol dapat mengatur suhu disekitar bioreaktor diperlukan juga sistem aktuator heater berupa kawat nikelin yang mampu mengubah besaran listrik menjadi besaran kalor melalui *duty cycle* tegangan suplai catu daya yang diatur oleh sinyal pwm mikrokontroller terhadap *gate* transistor MOSFET IRF540N sehingga pada akhirnya akan mengatur besarnya arus yang melewati kawat nikelin dan akan menimbulkan *drop* tegangan dimana diakibatkan oleh arus listrik melewati kawat dikonversikan menjadi panas akibat resistansi material nikelin.

### 1.5.5 Pengujian Keseluruhan Sistem Kontrol

Pada tahap pengujian dilakukan secara bertahap pada setiap bagian komponen penyusun sistem guna mengidentifikasi setiap permasalahan dan kekurangan pada sistem kontrol, terlebih permasalahan yang berpeluang merusak dan kinerja alat sehingga memastikan bahwa alat bekerja dengan sebagai mestinya. Adapun bagian pengujian yang duiji diantaranya ialah sistem sensor baik pH maupun suhu, sistem heater serta perangkat lunak kontroler logika fuzzy.

### 1.5.6 Analisa dan Evaluasi Sistem Kontrol

Tahapan selanjutnya setelah melakukan pengujian pada seluruh bagian sistem kontrol maka langkah selanjutnya ialah analisa. Analisa dilakukan bertujuan untuk mengetahui karakteristik perangkat lunak dan perangkat keras dari sistem kontrol. Apabila karakteristik sistem tersebut belum sesuai, maka perlu dilakukan evaluasi pada sistem untuk dirancang dan diuji kembali. Selain itu juga untuk mengevaluasi kualitas sampel rusip yang dihasilkan oleh sistem kontrol diperlukan evaluasi secara *human sensory test* kepada penikmat rusip melalui pemberian kuisioner.

### 1.5.7 Penyusunan Buku Tugas Akhir

Penyusunan buku tugas akhir akan dilakukan seiring dengan tahap-tahap lainnya., laporan terkait hasil proses yang telah dilakukan pada penelitian ini akan disusun dan disajikan dalam bentuk buku. Isinya meliputi pendahuluan, studi literatur, tinjauan pustaka, perancangan dan pembuatan sistem, pengujian dan analisa, serta penutup.

### 1.5.8 Penyusunan Jurnal Pomits

Penyusunan Jurnal POMITS akan dilakukan setelah penyusunan buku tugas akhir telah selesai.

## 1.6 Sistematika Penulisan

**BAB 1 PENDAHULUAN**

Bab ini menjelaskan mengenai latar belakang, permasalahan, batasan masalah, tujuan, metodologi, sistematika penulisan, serta relevansi dari penelitian.

**BAB 2 TINJAUAN PUSTAKA**

Dalam bab ini berisi mengenai teori dasar yang mendukung penyusunan buku tugas akhir, khususnya teori yang berhubungan dengan konsep dasar dari komponen yang digunakan.

**BAB 3 PERANCANGAN SISTEM**

Pada bab ini berisi perencanaan sistem baik itu perangkat keras (*hardware*) maupun perangkat lunak (*software*) dalam pembuatan alat ini.

**BAB 4 PENGUJIAN DAN ANALISIS**

Dalam bab ini menguraikan tetang pengujian dan analisis data dimana data tersebut diperoleh selama proses pengujian alat dan analisa atas hasil data tersebut.

**BAB 5 KESIMPULAN**

Bab ini berisi tentang kesimpulan terkait hasil yang telah dicapai pada penelitian yang telah dilakukan serta saran untuk pengembangan di masa mendatang.

## 1.7 Relevansi

Sistem Kontrol Suhu Pada Fermentasi Rusip Menggunakan Metode Logika Fuzzy Berdasarkan Tingkat Keasaman ini merupakan terobosan inovasi yang dapat membantu industri UMKM khususnya yang bergerak memproduksi hasil olahan fermentasi rusip dalam mempercepat waktu fermentasi dengan tetap menjaga kualitas akhir produk yang seragam satu sama lain, dimana dengan sistem kontrol melalui pemanas kawat nikelin ini mampu menjaga suhu bioreaktor tetap dalam keadaan suhu yang optimal bagi pertumbuhan bakteri asam laktat dibandingkan bakteri pathogen dengan penggunaan daya

listrik yang lebih efisien karena memiliki karakteristik konversi energi listrik menjadi kalor dengan penggunaan daya yang kecil. Dengan penggunaan sensor suhu dan pH sistem akan mampu menghasilkan produk dengan kualitas yang relatif sama antar produk yang dihasilkan dan meminimalisir kegagalan proses fementasi.

*Halaman ini sengaja dikosongkan*

# BAB 2
# TINJAUAN PUSTAKA

Suatu penelitian memerlukan teori-teori yang telah ada sebelumnya untuk dikaji lebih dalam guna memperkuat argumen penulis. Teori tersebut digunakan oleh penulis sebagai pedoman dasar dalam perancangan suatu penelitian.

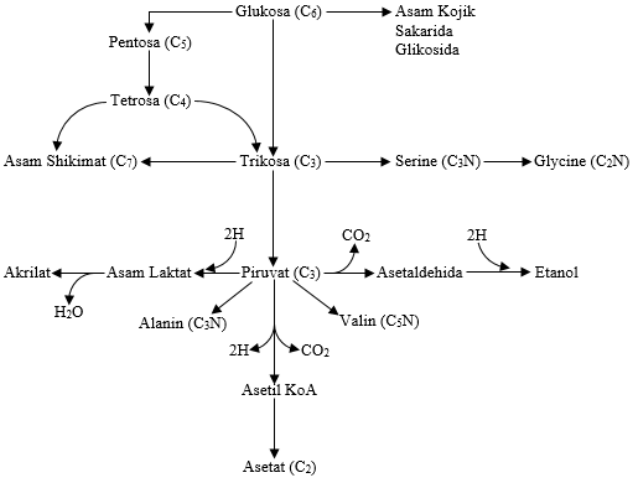
Pada bab ini akan berisi mengenai teori dasar yang menjadi landasan untuk merumuskan dan menyelesaikan masalah yang akan dibahas pada penelitian ini. Pada bagian ini terdapat beberapa tinjauan pustaka tentang setiap komponen yang akan digunakan secara umum dalam pembuatan alat pada penelitian ini.

## 2.1 Fermentasi

Kata fermentasi berasal dari Bahasa latin 'Fermentare', yang berarti "untuk ditinggalkan". Sedangkan definisi luas untuk praktek fermentasi adalah pentransformasian makanan oleh berbagai macam bakteri, fungi, dan enzim yang mereka hasilkan [6]. Fermentasi sudah lama ditemukan dalam berbagai macam kebudayaan manusia di seluruh dunia sejak ribuan tahun lalu. Jejak proses fermentasi tertua dapat ditemukan dalam bentuk anggur dan bir di Babylonia yang diperkirakan sudah ada sejak 7000 tahun sebelum masehi. Pada mulanya proses fermentasi ditemukan sebagai cara pengawetan makanan, namun dikarenakan adanya kemajuan teknologi yang pesat, mengakibatkan munculnya berbagai alat pengawetan makanan jenis lain seperti kulkas, pengkalengan makanan, dan *vacuum sealing*. Proses fermentasi belakangan ini lebih cenderung banyak digunakan untuk meningkatkan rasa dari makanan [7].

Fermentasi berkaitan erat pada olahan makanan hasil laut atau '*Seafood'* yang dapat digolongkan menjadi dua jenis fermentasi berdasarkan pengolahannya yakni fermentasi alami dan fermentasi terkontrol dimana keduanya memanfaatkan agen fermentasi seperti bakteri asam laktat ditambahkan sebagai unsur eksternal. Proses fermentasi pada ikan dapat dibagi menjadi proses basah (tanpa pengeringan), dan proses kering (dengan pengeringan). Berdasarkan proporsi kandungan garam, proses fermentasi pada ikan dapat dibedakan menjadi proses fermentasi bergaram tinggi apabila melebihi 20% dari berat total, bergaram rendah (6-8%), dan tidak

bergaram. Pemberian garam sangatlah penting pada proses pengawetan ikan, baik itu proses basah maupun kering karena dengan pemberian garam, air akan dikeluarkan dari daging ikan melalui proses osmosis, sehingga akan mengurangi kadar air sebagai bahan untuk perkembangan bakteri yang tidak dibutuhkan terutama bakteri pembusukan. Pemberian garam juga dapat menghambat beberapa macam mikroorganisme terutama bakteri pathogen pembusuk makanan dan enzim, sehingga kadar garam juga dapat digunakan untuk menentukan bakteri yang diperbolehkan untuk berkembang.




**Gambar 2. 1.** Diagram Pemecahan Molekul Glukosa Menjadi Molekul Asam Laktat Pada Proses Fermentasi Bakteri Asam Laktat [8]

Pada gambar 2.1, gambar tersebut menunjukan bagaimana tahapan-tahapan penguraian selama proses fermentasi bakteri asam laktat berlangsung yang menguraikan senyawa glukosa menjadi asam laktat. Terlihat bahwa proses fermentasi homolaktat menghasilkan laktat sebagai produk akhir yang dihasilkan melalui 2 mol piruvat yang berasal dari hasil turunan pemecahan kedua dari molekul glukosa.

Secara keseluruhan pemecahan glukosa pada fermentasi bakteri asam laktat terbagi menjadi dua kategori ditinjau dari jenis

bakteri yang menjadi agen pengurai, yaitu fermentasi homolaktat dan heterolaktat. Pada fermentasi homolaktat, satu molekul glukosa dikonversikan menjadi dua molekul asam laktat dapat dilihat pada persamaan (2.1) sebagai berikut:

$$C_6H_{12}O_6 \rightarrow 2CH_3CHOHCOOH \quad (2.1)$$

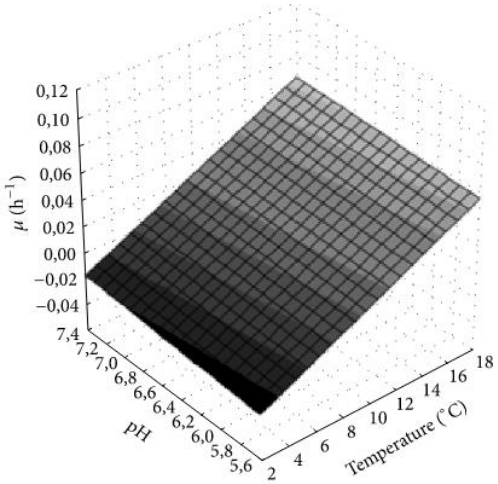
Sedangkan pada fermentasi heterolaktat, terjadi reaksi sebagai berikut, dengan satu molekul glukosa dikonversikan menjadi satu molekul asam laktat, satu molekul ethanol, dan satu molekul karbondioksida dapat dilihat pada persamaan (2.2) sebagai berikut:

$$C_6H_{12}O_6 \rightarrow CH_3CHOHCOOH + C_2H_5OH + CO_2 \quad (2.2)$$

Proses fermentasi dapat dipercepat dengan cara mengatur suhu lingkungan bakteri sesuai pada tingkatan kadar pH, Peningkatan suhu dapat memicu pertumbuhan bakteri dan secara beriringan pula meningkatkan produksi asam laktat yang semakin lama membuat produk fermentasi semakin bersifat asam secara grafik hal ini dapat dilihat dari gambar 2.2 dimana menjelaskan pengaruh kadar pH dan suhu terhadap pertumbuhan bakteri asam laktat:

## 2.2 Rusip

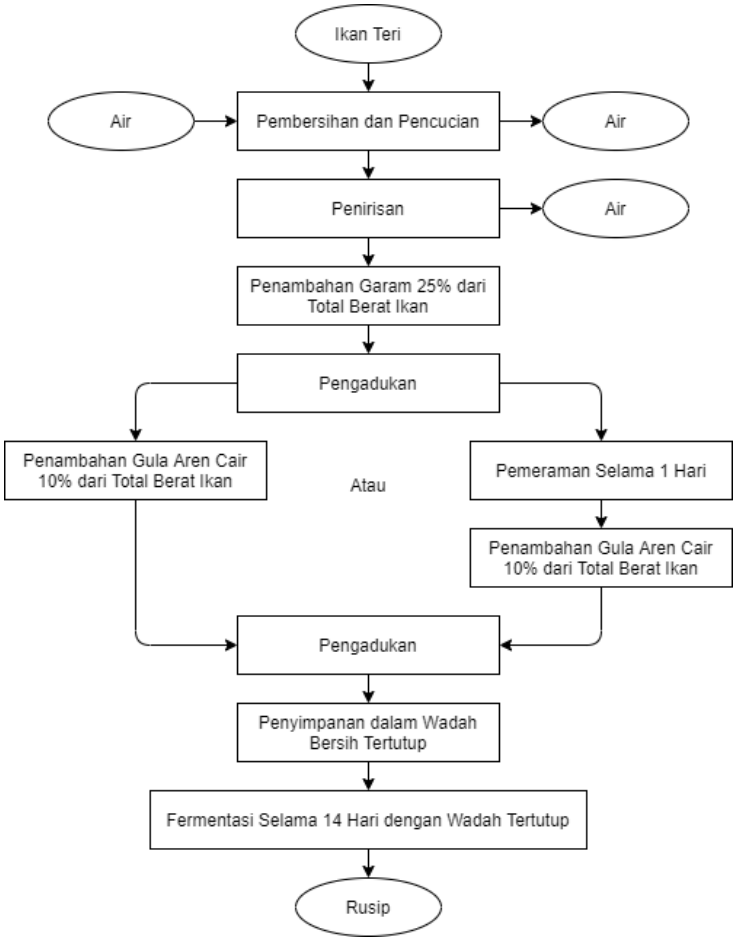
Rusip merupakan salah satu makanan olahan tradisional yang memanfaatkan fermentasi bakteri asam laktat. Rusip dapat dibuat dari bahan baku ikan yang difermentasi menggunakan garam dan gula [9]. Umumnya ikan teri atau ikan bilis (Stolephorus.sp) dipilih sebagai bahan utama rusip haruslah dalam keadaan segar. Karenanya, agar ikan terhindar dari proses pembusukan diperlukanlah penambahan garam. Penggunaan garam bertujuan sebagai media seleksi penekanan populasi bakteri pathogen yang merugikan sehingga bakteri asam laktat menjadi bakteri mayoritas pada awal proses fermentasi rusip. Sedangkan penggunaan gula bertujuan untuk pemasokan sumber energi bagi bakteri asam laktat dan penekanan jumlah kapang yang terkandung dalam rusip. Lama waktu fermentasi rusip berkisar 2 minggu dengan jenis fermentasi anaerob (tanpa membutuhkan oksigen). Bentuk fisik dari produk rusip dalam kemasan dapat dilihat pada gambar 2.3.




**Gambar 2. 2.** Grafik Pengaruh Pengingkatan Temperatur dan pH terhadap Pertumbuhan Bakteri Asam Laktat [9]

Rusip yang sudah melewati tahap fermentasi memiliki ciri-ciri khusus secara penilaian pendekatan indra-indra manusia mulai dari indra penglihatan berbentuk daging ikan utuh, empuk, cerah, dengan terdapat cairan keruh berwarna kecoklatan keabu-abuan hingga coklat gelap disekelilingnya, apabila dinilai indra penciuman akan berbau keasaman yang segar, sedangkan secara penilaian citra rasa memiliki rasa gurih, asin dan asam seimbang. Untuk langkah-langkah pembuatan rusip dapat dilihat pada gambar 2.3, sedangkan kandungan dan karakteristik terkandung dalam rusip dapat diamati pada tabel 2.1.

**Gambar 2. 3.** Rusip Dalam Kemasan [10]




**Gambar 2. 4.** Proses Pembuatan Fermentasi Rusip [12]

## 2.3 pH (*Puissance d'Hydrogen*)

Derajat Keasaman atau pH dan derajat alkalinitas atau pOH dari suatu media cairan, bergantung pada konsentrasi ion hidrogen ($H^+$) dan hidroksil ($OH^-$) [13]. Dalam usaha menghindari kesulitan penulisan yang muncul ketika nilai pH disajikan dalam bentuk faktor 10 berpangkat negatif seperti $10^{-7}$. Maka Sorensen menyusun konsep

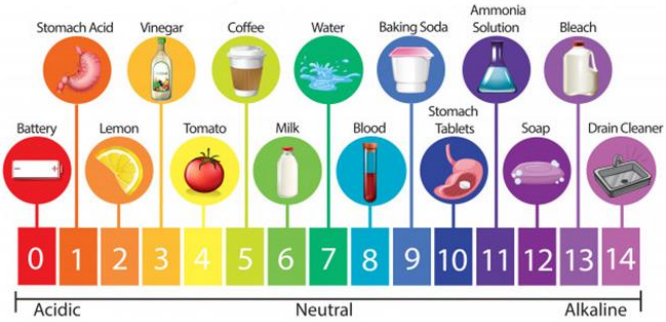
satuan unit pH yang mudah untuk diaplikasikan, kekuatan kadar hidrogen (*puissance d'hydrogen*), atau pOH (*puissance d'hydroxyl*) di definisikan melalui bentuk logaritma berbasis 10 dari konsentrasi ion hidrogen dimana dapat dilihat pada persamaan (2.3) sebagai berikut:

$$pH = \log\frac{1}{(H^+)} = -\log(H^+) \qquad (2.3)$$

Karena parameter pH bersifat logaritmik, yang mengartikan bahwa perubahan 1 satuan pH sesuai dengan perubahan sepuluh kali lipat dalam konsentrasi ion $H^+$. berdasarkan persamaan 2, nilai pH akan bernilai negatif logaritma dari ion hidrogen atau positif logaritma dari ion hidroksa. Dari segi penyajian skala pH sering dikatakan berkisar antara 0 hingga 14, dan sebagian besar sesuai untuk berbagai aplikasi pengukuran, meskipun masih terdapat kemungkinan untuk mendapatkan pH di bawah 0 ataupun di atas 14. Suatu cairan dikatakan bersifat asam ketika bernilai lebih rendah dari 7.0, dan kebalikannya suatu cairan dikatakan bersifat basa ketika bernilai lebih tinggi dari 7.0.

**Tabel 2. 1.** Karakteristik Kandungan Rusip [11]

| Parameter | Rusip Komersial*) | Rusip Hasil Laboratorium**) | Rusip Hasil Laboratorium Ditambah Starter *P. acidilactici****) |
|---|---|---|---|
| Kadar air (%) | 62,19—83,74 | 62,49—66,62 | 70,55 |
| Kadar protein (%) | 10,52—14,45 | 16,43—16,71 | - |
| Kadar lemak (%) | 1,82—3,06 | 0,71—0,90 | - |
| Kadar abu (%) | - | 9,23—14,41 | - |
| Kadar garam (%) | 17—30 | 3,95—7,90 | 10,18 |
| Kadar asam laktat (%) | - | 2,45—3,19 | 1,19 |
| pH | 5,01—6,10 | 4,56—4,33 | 5,83 |
| TVN (mg N/100g) | 1,65—2.384,54 | - | 40 |
| TMA (mg N/100g) | 11,55—94,58 | - | 3,36 |




**Gambar 2. 5. Skala pH [14]**

## 2.4 PH Meter

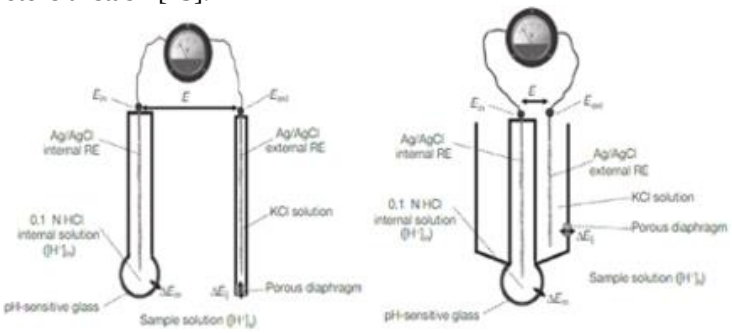
PH meter merupakan perangkat elektronika yang digunakan untuk mengukur tingkat keasaman cairan. Bagian utama dari pH meter tidak lain adalah probe pengukur seperti elektroda gelas, namun untuk pengaplikasian khusus, ISFET atau ion selective field-effect transistor lebih cenderung untuk dipilih. Seluruh pH meter sebelum digunakan harus dilakukan kalibrasi pada larutan buffer pH yang telah diketahui nilainya.

Jenis pH meter atau sensor pH yang mayoritas digunakan secara luas berbasis elektroda kaca. Elektroda kaca termasuk kelompok keluarga elektroda ion selektif, dimana elektroda kaca ini mengandung material yang sensitif terhadap perubahan konsentrasi ion terlarut dalam cairan pengujian. Dengan cara ini, besar drop tegangan antara elektroda dan permukaan terluar material sepenuhnya bergantung pada konsentrasi saja. Struktur elektroda kaca dalam pengukuran pH tersusun atas membran kaca berbentuk bolam terdiri 3 buah komposisi bahan yaitu 72% SiO2 - 22% Na2O - 6% CaO atau 80% SiO2 - 10% Li2O – 10% CaO.

Interaksi proton dalam cairan pengujian dengan permukaan elektroda kaca berbasis pada pertukaran sifat ion, ketika komponen silika utama bersentuhan dengan air, elektroda kaca terhidrasi dan berdisosiasi menjadi Si-O- dan pada gilirannya dapat terprotonasi sebagian dengan reaksi kimia pada persamaan (2.4) adalah sebagai berikut

$$\text{Si-}O^- + \text{H3}O^+ \equiv \text{Si-O-}H^+ + \text{H2O} \qquad (2.4)$$

Seringkali elektroda kaca dikemas dalam bentuk bulat, umumnya terbuat dari gelas bening atau tabung plastik dilengkapi membran kaca khusus yang sensitif terhadap perubahan kadar pH setebal 0.1 mm. Untuk gambaran sensor pH secara spesifik dapat dilihat pada gambar 6. Pada bagian dalam Elektroda kaca terdapat larutan HCl 0.1 N (terkadang dengan tambahan cairan AgCl jenuh), dan elektroda referensi internal (umumnya berbentuk kawat elektroda dengan lapisan garam Ag/AgCl) [13]. Konstannya jumlah larutan HCl pada sistem ini membuat nilai electrode Ag/AgCl memiliki nilai potensial stabil [15].




**Gambar 2. 6.** Skematik Representasi pH Meter Terdiri Dari Elektroda Kaca Terkopel Dengan Elektroda Referensi Terpisah (a) Dan Sistem Gabungan (b) [13]

**Tabel 2. 2.** Spesifikasi Elektroda Kaca

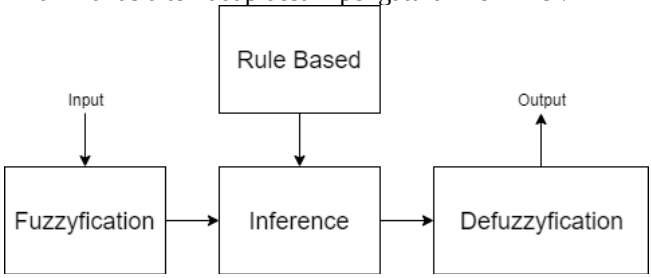
| **Probe pH meter** | Elektroda Kaca |
|---|---|
| **Rentang pH** | 0 – 14PH |
| **Rentang temperature** | 0 – 80ºC |
| **Titik-nol** | 7 ± 0.25PH |
| **Kesalahan alkali** | ≤15mV |
| **Persentase kemiringan teoritis** | ≥98% |
| **Resistansi internal** | ≤250MΩ |
| ***Response time*** | ≤1menit |
| **Temperatur operasi** | 0 – 60ºC |
| **Blok terminal** | BNC *plug* |

**Tabel 2. 3.** Spesifikasi Modul pH Meter

| | |
|---|---|
| **Modul** | pH-4052C |
| **Tegangan operasi** | 5V ± 0.2V (AC DC) |
| **Arus operasi** | 5 – 10mA |
| **Rentang deteksi konsentrasi pH** | 0 – 14 |
| **Rentang deteksi suhu** | 0 – 80ºC |
| ***Response time*** | ≤5S |
| ***Settling time*** | ≤60S |
| **Disipasi daya komponen** | ≤0.5W |
| **Temperatur operasi** | -10 – 50ºC (temperatur nominal 20ºC) |
| **Kelembaban** | 95% RH (kelembaban nominal 65% RH) |
| **Output** | Sinyal tegangan analog |

## 2.5 Logika Fuzzy

Logika Fuzzy merupakan salah satu cabang ilmu Artificial Inteligence, yaitu suatu yang membuat komputer dapat meniru kecerdasan manusia [16]. Logika fuzzy tergolong mudah diaplikasikan dan dimengerti oleh pengguna karena menyediakan solusi terhadap suatu permasalahan kompleks dalam berbagai ranah bidang kehidupan yang dijawab dengan pendekatan penalaran dan pengambilan keputusan layaknya manusia. Oleh sebab itu ketika logika fuzzy diterapkan ke dalam suatu kontroler, dimana seringkali disebut sebagai kontrol yang menerapkan pendekatan heuristik yang dengan mudah menanamkan pengetahuan dan elemen kunci dari pemikiran manusia terhadap desain pengaturan nonlinier.



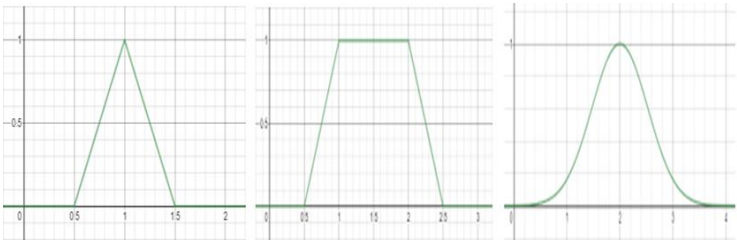

**Gambar 2. 7.** Arsitektur Logika Fuzzy [17]

Bentuk arsitektur logika fuzzy yang ditunjukan pada gambar 2.7, umumnya terdiri dari 4 buah komponen utama yaitu *Rule Base*, Fuzzifikasi, *Inference Engine*, dan Defuzzifikasi. Pada bagian Rule base biasanya tersusun atas seluruh aturan yang dibentuk sesuai dengan pendapat ahli untuk mengontrol sistem pengambilan keputusan secara tepat. Selanjutnya, tahapan kedua pada awal langkah logika fuzzy adalah fuzzifikasi, yang mana bertugas mengubah crisp input menjadi variable linguistic dengan menggunakan fungsi keanggotaan yang telah diatur. Fungsi keanggotaan dapat dibentuk dari berbagai jenis, fungsi segitiga rumus, fungsi tapesium, atau fungsi gauss, masing-masing persamaan dapat dilihat secara berurutan pada persamaan (2.5), (2.6), dan (2.7) yang memiliki rumus sebagai berikut:

$$\mu_A(x) = \begin{cases} 0, & x \le a \\ \dfrac{x-a}{m-a}, & a < x \le m \\ \dfrac{b-x}{b-m}, & m < x < b \\ 0, & x \ge b \end{cases} \tag{2.5}$$

$$\mu_A(x) = \begin{cases} 0, & (x < a)\ or\ (x > b) \\ \dfrac{x-a}{m-a}, & a < x \le m \\ 1, & m \le x \le n \\ \dfrac{b-x}{b-n}, & n \le x \le b \end{cases} \tag{2.6}$$

$$\mu_A(x) = e^{-\frac{(x-m)^2}{2k^2}} \tag{2.7}$$



**Gambar 2. 8.** Fungsi Keanggotaan Logika Fuzzy Segitiga, Trapesium dan Gaussian [18]

Pada tahapan kedua, setelah input crisp diubah menjadi himpunan fuzzy, dilakukanlah *fuzzy inference*. Inferensi atau penyimpulan fuzzy merupakan proses merumusan pemetaan dari input yang diberikan ke output menggunakan logika fuzzy. Dari pemetaan tersebut memberikan dasar dari mana keputusan dapat dibentuk berdasarkan pola yang terlihat dari hasil penerapan Rule Base. Proses Inferensi Fuzzy melibatkan semua bagian seperti, fungsi keanggotaan, operator logika fuzzy, dan aturan jika-maka (*If-Then*). Konsep dari penggunaan aturan jika-maka (*If-Then*) ditampilkan pada persamaan (2.8).

$$If\ X_1\ is\ A_{i,1}\ \theta\ X_2\ is\ A_{i,2}\ \theta\ \dots \theta\ X_k\ is\ A_{i,k}\ Then\ Y\ is\ B \qquad (2.8)$$

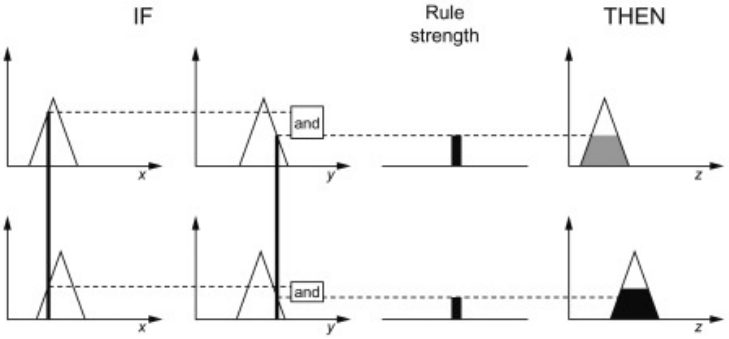
Terakhir pada tahapan ketiga, setelah tahap inference akan dilewatkan pada langkah defuzzyfikasi. Defuzzyfikasi sendiri merupakan merupakan proses konversi atau penggabungan untuk mendapatkan nilai crisp keluaran terhadap plant dari output himpunan fuzzy yang berhasil teragregasi oleh mekanisme inferensi sebelumnya [19]-[21]. Ada beberapa metode dalam literatur untuk melakukan defuzzyfikasi. Diantara semua metode yang ada, metode *center of gravity* (COG) termasuk metode yang paling diminati, secara matematis ditunjukkan dalam persamaan (2.9).

$$x_0 = \frac{\sum_{x=a}^{b} \mu_A\,(x)x}{\sum_{x=a}^{b} \mu_A(x)x} \qquad (2.9)$$

Dimana $\mu_A\,(x)$ adalah derajat fungsi keanggotaan output dan x adalah nilai crisp output.

Secara penggunaan luas, terdapat dua jenis bentuk pengimplementasian inferensi fuzzy: Tipe Madani (1977) dan Tipe Sugeno (1985). Akan tetapi di dalam penelitian ini dipilih inferensi tipe Mamdani. Inferensi tipe Mamdani dipertimbangkan sebagai metode yang paling sering diimplementasikan dalam berbagai sistem pengambilan keputusan fuzzy terhadap suatu permasalahan. Kelebihan metode logika fuzzy Mamdani dibandingkan Sugeno terletak pada kemampuan kekuatan ekspresif dan impretabilitas output Mamdani memanfaatkan penggunaan aturan bersifat fuzzy sesuai pendapat para ahli, sebaliknya output Sugeno yang didapatkan

cenderung mengarah pada pendekatan matematis berupa pemberatan rata-rata (*weighted average*) [22].




**Gambar 2. 9.** Proses Inferensi Fuzzy Tipe Madani [23]

**Gambar 2. 10.** Metode Defuzzifikasi Madani [24]

## 2.6 Sensor Temperatur DS18B20

Sensor temperature merupakan sebuah sensor yang dapat menangkap perubahan temperatur lingkungan lalu kemudian mengonversinya menjadi besaran nilai listrik [25]. Pada umumnya sensor temperature yang ada di pasaran memerlukan ADC dan penyambungan beberapa sambungan tambahan pada beberapa port mikrokontroler, Namun sensor DS18B20 ini memiliki keunggulan bila dibandingkan dengan sensor temperatur lain dimana secara konektivitas dengan mikrokontroler tidak memerlukan ADC melainkan hanya menggunakan 1 kabel komunikasi data (1 *wire* saja). DS18B20 merupakan sensor temperatur hasil produksi Dallas Semiconductor yang dilengkapi dengan fitur pengemasan waterproof,

sehingga membuat sensor tahan terhadap air dan dapat melakukan penangkapan perubahan temperatur di dalam suatu larutan.

**Gambar 2. 11.** Konfigurasi Sensor Temperatur DS18B20 [26]

Adapun konfigurasi pin dari sensor DS18B20 pada gambar 2.12 terdiri dari 3 buah yaitu vcc, gnd, dan data. Pin Vdd merupakan pin pemasukan tegangan sumber, pin data merupakan pin pengambilan output pengukuran temperatur oleh sensor, dan pin gnd merupakan pin yang terhubung dengan ground pada mikrokontroler serta spesifikasi singkat kemampuan dan pengoperasian dari sensor temperatur DS18B20 dapat dibaca pada tabel 2.4.

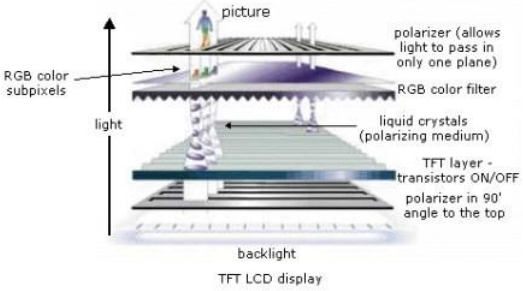
**Tabel 2. 4.** Spesifikasi Sensor Temperatur DS18B20

| | |
|---|---|
| **IC Sensor** | DS18B20 |
| **Tegangan operasi** | 3.3 – 5V |
| **Rentang suhu operasi** | -55 – 125ºC |
| **Kesalahan thermometer** | -10 – 85ºC (±0.5)<br>-30 – 100ºC (±1)<br>-55 – 125ºC (±2) |
| **Pengaturan resolusi** | 9-12bit |
| **Waktu konversi maksimum** | 750ms |
| **Output** | 1-*wire* |

## 2.7 Liquid Crystal Display Thin Film Transistor (LCD TFT)

TFT atau *Thin Film Transistor* merupakan perangkat semikonduktor yang digunakan untuk memperkuat dan mengubah sinyal elektronik berbasis film tipis dan lapisan dielektrik yang memiliki sifat anti-listrik serta elemen kimia pada lapisan penyelubungnya, yakni layar LCD. Sedangkan LCD atau *Liquid Crystal Display* merupakan salah satu jenis tampilan elektronik dirancang dengan teknologi gerbang logika CMOS yang bekerja menyalurkan cahaya tampilan ke layar dari *back-lit* atau *front-lit* ketika tidak diberikan tegangan, sebaliknya ketika diberikan tegangan maka akan membelokkan cahaya tampilan layar [27]. Secara susunan material, LCD tergolong sebagai lapisan dari campuran organic antara lapisan kaca bening dengan elektroda transparan indium oksida dalam bentuk tampilan seven segment maupun dapat diletakkan dalam lapisan elektroda pada kaca belakang.

Layar LCD TFT dapat digunakan untuk menampilkan output pemrosesan mikrokontroler berupa karakter, angka, huruf, bahkan grafik, serta dapat diatur melalui input pemberian perintah eksekusi pada area tertentu pada layar yang bersifat *resistive screen*. Teknologi Resistive screen sendiri tersusun atas beberapa lapisan. Ketika layer ditekan, lapisan terluar terdorong menuju lapisan selanjutnya. Teknologi ini akan bertindak sesuai tekanan yang diterapkan dan input yang didaftarkan melalui pemrograman yang diberikan. Oleh karena itu Layar LCD TFT termasuk sebagai layar LCD yang sering dioperasikan sebagai *interface* perantara manusia dengan perangkat yang ingin dikontrol.




**Gambar 2. 12.** Lapisan Penyusun LCD TFT [28]

## 2.8 Arduino Uno

Arduino Uno adalah salah satu mikrokontroler yang dikembangkan oleh Hernando Barragan, Massimo Banzi, David Cuartielles dan pendiri lainnya, tergolong sebagai papan mikrokontroler yang bersifat *open-source* berbasis ATmega328 sehingga memudahkan pengguna untuk mengembangkan dan memahami prinsip kerja perangkat tersebut. Papan mikrokontroler Arduino Uno sendiri tergolong memiliki konektivitas serta kemampuan terbilang cukup memumpuni untuk penerapan diberbagai proyek elektronika dimana perangkat Arduino uno dilengkapi dengan 14 *input*/*output* digital, 6 pin diantaranya dapat dioperasikan sebagai PWM *output* terletak pada pin 3, 5, 6, 9, 10, dan 11, 6 input analog, resonator keramik 16 MHz, koneksi USB, colokan suplai daya, *header* ICSP dan tombol *reset* sesuai ilustrasi gambar 2.13. Karena itu tipe Uno merupakan jenis Arduino yang tergolong paling sering digunakan dan terdokumentasi lengkap dari seluruh keluarga Arduino [29].

**Gambar 2. 13.** Arduino Uno [29]

Adapun penjelasan singkat mengenai spesifikasi mikrokontroler Arduino Uno mulai dari tegangan operasi, tegangan input, jumlah pin digital I/O, arus DC maksimum per pin, kapasitas *flash memory*, SRAM, EEPROM dan kecepatan *clock* dapat diamati pada tabel 2.5 sebagai berikut:

**Tabel 2. 5.** Spesifikasi Arduino Uno

| | |
|---|---|
| **Mikrokontroler** | ATmega328 |
| **Tegangan operasi** | 5V |
| **Tegangan *input* (rekomendasi)** | 7 – 12V |
| **Tegangan *input* (limit)** | 6 – 20V |
| **Pin I/O digital** | 14 (6 diantaranya menyediakan keluaran PWM) |
| **Pin *input* digital** | 6 |
| **Arus DC per pin I/O** | 40 mA |
| **Arus DC untuk pin 3.3V** | 50 mA |
| ***Flash memory*** | 32 KB, 0.5 KB digunakan oleh *bootloader* |
| **SRAM** | 2 KB |
| **EEPROM** | 1 KB |
| **Kecepatan *clock*** | 16 MHz |

### 2.8.1 Sumber Daya

Papan Arduino Uno dapat diaktifkan dengan memberikan suplai tegangan melalui koneksi USB atau suplai daya eksternal. Suplai daya dipilih dengan otomatis. Daya eksternal atau *non*-USB juga bisa dari adaptor AC-ke-DC atau baterai. Adaptor dapat dihubungkan dengan menancapkan colokan ukuran 2.1mm sumber positif ke colokan listrik pada papan. Petunjuk sambungan dari baterai dapat dimasukkan ke dalam *header* pin GND dan Vin dari konektor daya. Papan dapat dioperasikan dengan suplai eksternal 6 hingga 20 volt. Namun, jika disuplai dengan daya kurang dari 7V, pin 5V akan menghasilkan tegangan kurang dari 5V dan akan membuat Arduino Uno menjadi tidak stabil. Sebaliknya Ketika diberi tegangan eksternal lebih dari 12V, regulator tegangan yang terdapat pada Arduino Uno akan terlalu panas dan merusak papan. direkomendasikan pengoperasian mikrokontroler Arduino Uno pada kisaran tegangan 7-12V. Berikut *power pin* yang terdapat di Arduino Uno:

- VIN merupakan pin tegangan masukan Arduino jika tegangan eksternal digunakan. Selain melalui pin ini, tegangan eksternal dapat diberikan melalui *power jack*.
- 5V merupakan *pin output* dari tegangan berasal dari pin VIN, DC *power jack*, atau USB yang dikonversi menjadi 5V oleh *voltage regulator*.
- 3V3 merupakan pin yang memiliki fungsi sama dengan pin 5V

namun keluaran tegangan yang dihasilkan yaitu 3,3 V.

- GND merupakan pin *ground*

Mengenai detail informasi susunan diagram pin pada mikrokontroler Arduino Uno dapat diamati pada gambar 2.14.

### 2.8.2 Input dan Output

Pada Arduino Uno, terdapat 14 pin digital *input/output* dan 6 pin analog, Dalam penentuan pin digital digunakan sebagai pin input maupun output dapat diatur dengna penggunaan fungsi perintah seperti pinMode(), digitalWrite(), dan digitaRead(). Setiap pin digital tersebut bekerja pada tegangan pengoperasian maksimum 5V untuk kondisi HIGH dan 0V untuk kondisi LOW. Perlu diketauhi juga bahwa setiap pin tersebut hanya dapat menerima arus maksimum sebesar 40 mA, dan di dalamnya memiliki resistor *pull-up* internal sebesar 20-50 KOhms yang terputus pada kondisi normal. Dari 14 pin digital, beberapa memiliki fungsi spesifik yang dijabarkan sebagai berikut:

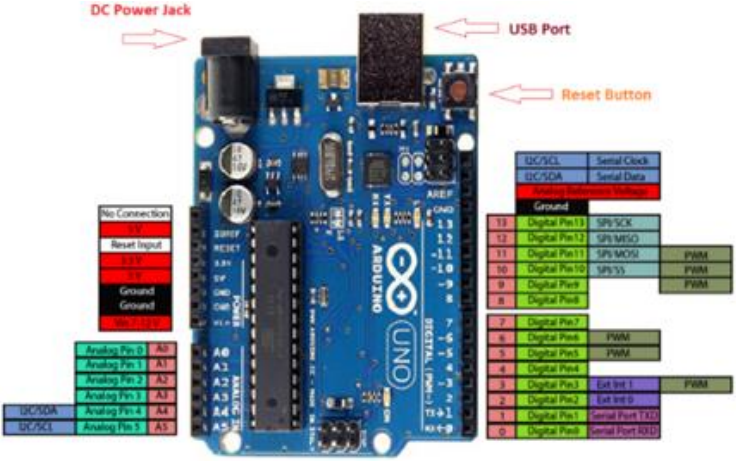
- Pin Serial 0(RX) dan 1(TX) digunakan sebagai komunikasi serial dimana pin 0 digunakan untuk menerima data serial TTL sedangkan pin 1 digunakan untuk mengirimkan data serial TTL.
- Pin *Interrupt* Eksternal 2 dan 3: Pin ini dapat dikonfigurasi untuk memicu interupsi pada keadaan logika LOW, *rising* atau *falling edge*, atau perubahan nilai.
- Pin 3, 5, 6, 10 dan 11 menyediakan 8-bit keluaran PWM dengan menggunakan fungsi analogWrite ().
- Pin 10(SS), 11(MOSI), 12(MISO), dan 13(SCK) digunakan untuk komunikasi SPI.
- Pin 13 terhubung dengan LED yang tertanam pada Arduino Uno, Ketika pin 13 berlogika HIGH maka LED akan menyala dan keita pin 13 berlogika LOW, LED akan mati.

  Terkhusus bagi pin analog pada Arduino Uno masing-masing pin dapat memberikan resolusi 10 bit, yakni mengeluarkan nilai dengan rentang 0 hingga 1024 dimana mengonversi tegangan dari 0V hingga 5V akan tetapi batas rentang nilai analog tersebut masih dapat diubah dengan menggunakan pin AREF dengan fungsi analogReference() [31]. Berikut penjabaran lebih lanjut dari penjelasan pin analog yang terdapat pada Arduino Uno:
- Pin Reset akan mereset mikrokontroler apabila diberi nilai LOW.

- Pin A4 dan A5 membantu komunikasi TWI/I2C diaktifkan menggunakan library Wire.
- pin AREF yang berfungsi sebagai tegangan referensi untuk masukan analog.

### 2.8.3 Pin Komunikasi

Arduino Uno menyediakan UART TTL (5V) komunikasi serial dengan menggunakan pin *digital* 0 dan pin 1 agar dapat berkomunikasi dengan sebuah komputer, Arduino lain, atau mikrokontroler lain. Pada papan Arduino Nano terdapat ATmega16U2 yang menghubungkan komunikasi serial melalui USB dan tampil sebagai *virtual* COM *port* pada Arduino *software* dalam komputer. Arduino *software* memiliki serial monitor yang memungkinkan data teks sederhana ke atau dari Arduino. LED TX dan RX akan menyala ketika data dikirimkan ke komputer. Selain komunikasi serial, ATmega328 juga mampu melakukan komunikasi I2C dan SPI [32].




**Gambar 2. 14.** Diagram Pin Arduino Uno [30]

### 2.8.4 Pemrograman

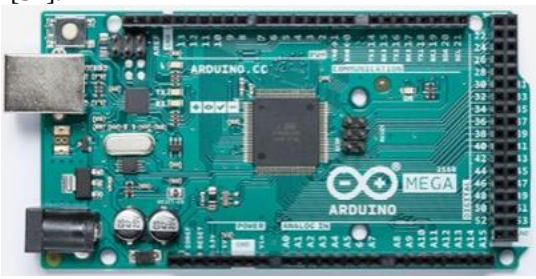
Arduino Uno dapat di program dengan Arduino Software. Sebelum meng-upload program ke Arduino Uno, terlebih dahulu pilih Arduino /Genuino Uno pada menu Tool > Board. ATmega328 pada

Arduino Nano telah dilengkapi dengan bootloader yang memungkinkan proses upload program baru ke Arduino Uno tanpa menggunakan pemrograman external hardware.

Sesudah membuat rangkaian yang diinginkan baik di breadboard maupun yang telah disolder, program harus diunggah (dikenal sebagai sketsa) ke Arduino. Sketsa adalah seperangkat instruksi yang memberi tahu kepada Arduino apa fungsi yang harus dilakukan. Arduino hanya dapat menahan dan melakukan satu sketsa pada suatu waktu. Perangkat lunak yang digunakan untuk membuat sketsa Arduino disebut IDE yang merupakan singkatan dari *Integrated Development Environment* [33].

## 2.9 Arduino Mega 2560

Arduino Mega 2560 merupakan salah satu jenis mikrokontroler Arduino berbasis ATmega2560. Mikrokontroler ini memiliki 54 pin input atau output digital, dimana 15 diantaranya dapat dioperasikan sebagai output PWM. Selain itu Arduino Mega 2560 dilengkapi 16 pin input analog, 4 UART atau port serial perangkat keras, kristal osilator 16 MHz, koneksi USB, colokan listrik, *header* ICSP, dan tombol reset, ditunjukkan pada gambar 2.15. Oleh karena itu, mikrokontroler ini telah terdesain secara lengkap, memenuhi semua yang diperlukan untuk mendukung pengunaan mikrokontroler. Dalam mengoperasikannya pengguna hanya perlu menyambungkan mikrokontroler ke komputer dengan kabel USB atau daya dengan adaptor AC ke DC atau baterai. Mikrokontroler ini juga kompatibel dengan Sebagian besar *shield* atau modul tambahan mikrokontroler yang dirancang untuk Uno dan papan sebelumnya Duemilanove atau Diecimila [34].




**Gambar 2. 15.** Arduino Mega 2560 [34]

Arduino mega 2560 merupakan versi terbaru pengganti Arduino Mega, keunggulan versi terbaru ini dibandingkan generasi sebelumnya, tidak lagi mengunakan *chip driver* FTDI USB to serial. Akan tetapi digantikan *chip* ATmega 16U2 atau ATmega 8U2 pada papan revisi 1 dan revisi 2 yang diprogram sebagai converter USB to serial. Arduino Mega 2560 revisi 2 ini memiliki resistor penarik jalur HWB 8U2 ke Ground, sehingga lebih mudah untuk dimasukkan ke dalam mode DFU. Selain itu sama halnya dengan Arduino Uno, mikrokontroler Arduino Mega 2560 mendukung bahasa pemrograman Arduino C yang dapat digunakan pada Arduino IDE. Untuk spesifikasi lengkap dari Arduino Mega dapat dilihat pada tabel 2.6 sebagai berikut:

**Tabel 2. 6.** Spesifikasi Arduino Mega 2560

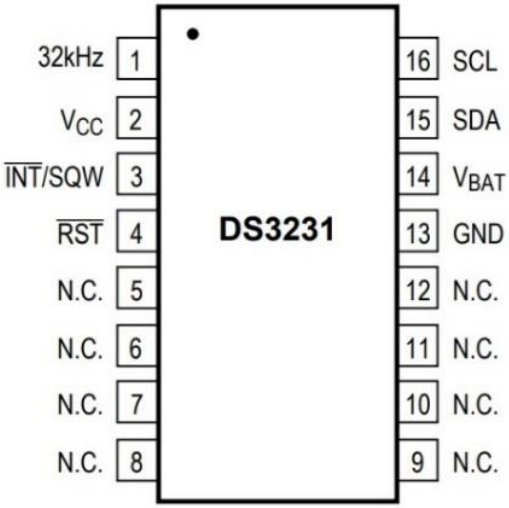
| | |
|---|---|
| **Mikrokontroler** | Atmega2560 |
| **Tegangan operasi** | 5V |
| **Tegangan *input* (rekomendasi)** | 7 – 12V |
| **Tegangan *input* (limit)** | 6 – 20V |
| **Pin I/O digital** | 54 (15 diantaranya menyediakan keluaran PWM) |
| **Pin *input* digital** | 16 |
| **Arus DC per pin I/O** | 40 mA |
| **Arus DC untuk pin 3.3V** | 50 mA |
| ***Flash memory*** | 256 KB, 8 KB digunakan oleh *bootloader* |
| **SRAM** | 8 KB |
| **EEPROM** | 4 KB |
| **Kecepatan *clock*** | 16 MHz |

Apabila dilihat secara seksama susunan pin diagram Arduino Mega pada gambar 2.19. Terlihat jelas pada gambar bahwa tidak banyak perbedaan antara Arduino uno dan Arduino Mega terkecuali spesifikasi ruang memori lebih besar, ukuran papan lebih besar

sehingga mampu menampung lebih banyak pin I/O dan pin komunikasi.

## 2.10 RTC DS3231

*Real Time Clock* atau RTC adalah sebuah perangkat untuk mengetahui waktu yang dibuat dalam bentuk rangkaian terintegrasi atau (IC). DS3231 sendiri merupakan sebuah RTC yang dikembangkan oleh Maxim Integrated yang ditawarkan dengan harga murah. DS3231 telah terintegrasi dengan osilator cristal yang sudah terkompensasi terhadap suhu, bekerja dengan protokol komunikasi I2C serta dilengkapi dengan baterai cadangan, dimana baterai cadangan tersebut berfungsi sebagai suplai cadangan apabila suplai daya utama mati. Kelebihan lain modul RTC DS3231 ialah kemampuannya menyimpan informasi detik, menit, jam, tanggal, bulan, dan tahunan yang telah dilengkapi fitur perubahan tanggal dan waktu pada akhir bulan secara otomatis termasuk koreksi untuk tahun kabisat. Detail pin diagram dari RTC DS3231 dapat dilihat pada gambar 2.16, sedangkan spesifikasi singkat komponen IC dan deskripsi pinnya secara turut-turut dapat dilihat pada tabel 2.7, dan 2.8.




**Gambar 2. 16.** Pin Diagram RTC DS3231 [36]

**Tabel 2. 7.** Spesifikasi RTC DS3231

| | |
|---|---|
| **Tegangan Masukan** | 3-5 VDC |
| **Dimensi** | 3,8 cm x 2,3 cm |
| **Suhu Maksimum Saat Beroperasi** | +85ºC |
| **Suhu Minimum Saat Beroperasi** | -40º |
| **Output** | Jam, Menit, Detik, Tanggal, Bulan, Tahun |

**Tabel 2. 8.** Deskripsi Pin RTC DS3231

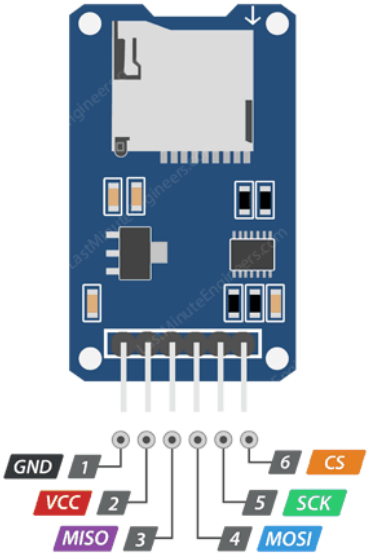
| Nomor Pin | Nama Pin | Deskripsi |
|---|---|---|
| 1 | 32KHz | 32KHz *Output* |
| 2 | Vcc | *DC Power Pin* |
| 3 | INT/SQW | *Active Low Interrupt or Square wave Ouput* |
| 4 | RST | *Active Low Reset* |
| 5-12 | NC | *No Connection* |
| 13 | GND | *Ground* |
| 14 | $V_{BAT}$ | *Back-up Power Supply Input from Battery* |
| 15 | SDA | *Serial Data I/O* |
| 16 | SCL | *Serial Clock Input* |

## 2.11 Modul Data Logger Kartu Mikro SD

Seringkali modul kartu mikro SD atau SD Card banyak digunakan sebagai media pengaplikasian *Data logging* ke dalam suatu alat. *Data Logger* sendiri merupakan suatu perangkat elektronik yang digunakan untuk menyimpan data dari waktu ke waktu, umumnya dikenal sebagai pencatatan data [37]. Dalam penerapannya *data logging* sudah banyak diimplementasikan di beberapa aplikasi seperti Cuaca (Suhu), Pertanian (Kelembaban Tanah), Mobil (Data Kecelakaan), Pesawat Udara (Black Box), dan lain-lainnya.

Kartu Micro SD sendiri merupakan perangkat memori berbasis flash yang dapat dilepas. Tergolong sebagai memori yang *non-volatile* dan sering digunakan di ponsel dan perangkat elektronik konsumen

lainnya. Umumnya setiap kartu Micro SD bekerja dalam kisaran tegangan 2,7V hingga 3,6V (biasanya, 3,3V) dan mendukung protokol komunikasi SPI. Oleh karena itu modul *data logging* kartu mikro SD terdiri atas dua bagian *level shifter* dan regulator tegangan. *Level shifter* berfungsi untuk mengubah logika *interface* atau antarmuka dari 5V ke 3.3V maupun sebaliknya, sedangkan komponen regulator berfungsi untuk mengubah tegangan mikrokontroler yang bertegangan 5V menjadi tegangan 3.3V sesuai dengan tegangan pengoperasian kartu mikro SD. Secara fisik dan susunan pin diagram dari modul kartu mikro SD dapat dilihat pada gambar 2.17.




**Gambar 2. 17.** Pin Diagram Modul Kartu Mikro SD [38]

**Tabel 2. 9.** Spesifikasi Modul Kartu Micro SD

| | |
|---|---|
| **Tegangan Masukan** | 3,3V atau 5 VDC |
| **Arus** | 0,2-200mA |
| **Dimensi** | 42 x 24 x 12 mm |
| **Mendukung Jenis Kartu** | Micro SD (2G)<br>SDHC (3G) |
| **Komunikasi Perangkat** | Serial Peripheral Interface (SPI) |

## 2.12 IRF540N

IRF540N merupakan transistor bertipe Metal Oxide Field Effect Transistor (MOSFET) yang terdiri atas 3 buah seperti divisualisasikan pada gambar 2.18. Dimana 3 buah pin tersebut adalah Gate, Drain, Source. Gate adalah pin pengatur pensaklaran jalannya arus listrik dari Drain ke Source, drain merupakan pin yang terhubung langsung dengan beban yang ingin diatur, sedangkan source merupakan pin yang terhubung dengan bagian *ground*. Tidak seperti transistor biasa atau *Bipolar Junction Transistor* (BJT), MOSFET merupakan perangkat komponen yang dikendalikan tegangan. Artinya, MOSFET dapat diaktifkan atau dimatikan oleh suplai sesuai tegangan ambang batas Gerbang (VGS) yang diperlukan. Selain itu karena IRF540N termasuk sebagai MOSFET N-Channel maka ketika tegangan gerbang diterapkan pin gate maka arus dari pin drain akan mengalir ke source, sedangkan ketika tidak ada tegangan yag diterapkan pada pin gate maka pin drain dan source akan dibiarkan dalam keadaan terbuka [39]. Perlu diketahui bahwa IRF540N sendiri memiliki tegangan *threshold* sebesar 4V, yang berarti dapat dengan mudah diatur oleh tegangan rendah seperti 5V dan umumnya MOSFET IRF540N memiliki spesifikasi kemampuan mengendalikan beban arus normal sebesar 23A dan mendukung arus puncak hingga 110A. Oleh sebab itu IRF540N banyak digunakan oleh mikrokontroler sebagai *logic switching*.

## 2.13 2N3904

2N3904 adalah transistor bertipe BJT dengan konfigurasi NPN sehingga kolektor dan emitor akan dibiarkan terbuka (*Reverse bias*) ketika pin dasar ditahan berlogika LOW atau terhubung dengan Ground maka transistor akan tertutup (*Forward bias*) ketika sinyal diberikan ke pin basis. 2N3904 memiliki nilai gain sebesar 300; nilai tersebut menentukan kapasitas amplifikasi transistor. Jumlah maksimum arus yang dapat mengalir melalui pin Kolektor adalah 200mA, karenanya 2N3904 tidak dapat dihubungkan ke beban yang mengonsumsi lebih dari 200mA. Sebagai saklar transistor 2N3904 harus dikendalikan oleh arus yang berada pada pin basis, kapasitas arus basis dibatasi hingga 5mA. Karenanya resistor selalu ditambahkan secara seri dengan pin dasar. Nilai resistor ini (RB) dapat dilihat pada persamaan (2.10) sebagai berikut:

$$RB = VBE / IB \tag{2.10}$$

Dimana, nilai VBE harus 5V untuk 2N3904 dan Arus basis (Ib) tergantung pada arus Kolektor (Ic). Berdasarkan lampiran *datasheet*, besar arus basis tidak boleh melebihi 5mA.

Selain itu ketika transistor ini terbiaskan secara penuh maka dapat mengalirkan arus maksimum sebesar 200mA melintasi kolektor dan emitor. Tahap ini disebut Daerah Saturasi dan tegangan khas yang diperbolehkan melintasi Collector-Emitter (VCE) atau Collector-Base (VCB) masing-masing dapat bertegangan hingga 40V dan 60V sedangkan ketika arus basis dilepas, transistor menjadi sepenuhnya mati, tahap ini disebut sebagai *cut-off region* dan tegangan *base-emitter* dapat mencapai 600 mV. Adapun bentuk fisik dan diagram pin dari transistor 2N3904 dapat dilihat pada gambar 2.20.

## 2.14 Kawat Nikrom

Kata Nikrom umumnya mengacu pada campuran material nikel, kromium, besi dan elemen lainnya. Campuran material Nichrome umumnya digunakan sebagai kawat resistansi. Variasi komposisi kawat nikrom paling banyak ditemui seringkali adalah nichrome-80 dan nichrome-60 yang masing-masing memiliki kadar kandungan nikel sekitar 80% dan 60% sedangkan kromium sekitar 20% dan 40%.

Kelebihan kawat nikrom dibandingkan jenis elemen pemanas lain diantaranya adalah ketahanan terhadap oksidasi, stabilitas pada suhu tinggi, dan resistivitas yang lebih tinggi daripada banyak logam lain membuat nichrome bahan yang baik untuk elemen pemanas listrik, terutama terdapat pada peralatan dan alat yang membutuhkan pemanasan. Biasanya, nikrom dililit dalam bentuk gulungan dengan hambatan listrik tertentu, dan ketika arus melewatinya untuk menghasilkan panas [41].

Secara lengkap pengaturan panas yang akan dihasilkan oleh kawat nikrom mulai dari besar resistansi kawat, perubahan resistansi kawat oleh pengaruh temperatur, hambatan jenis kawat, dan kenaikan suhu kawat masing-masing dapat dilihat pada-persamaan (2.11), (2.12), (2.13), dan (2.14) sebagai berikut:

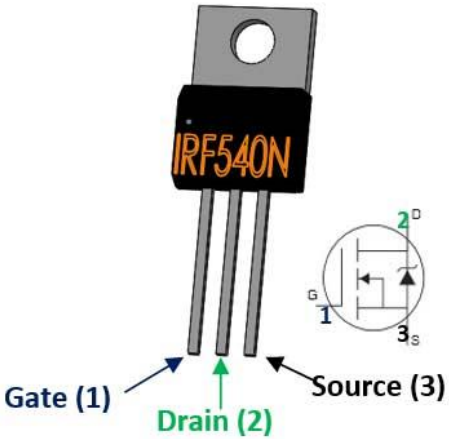
$$R = \frac{\rho * L}{A} \tag{2.11}$$

$$R = R_o(1 + \alpha(T - T_o)) \tag{2.12}$$

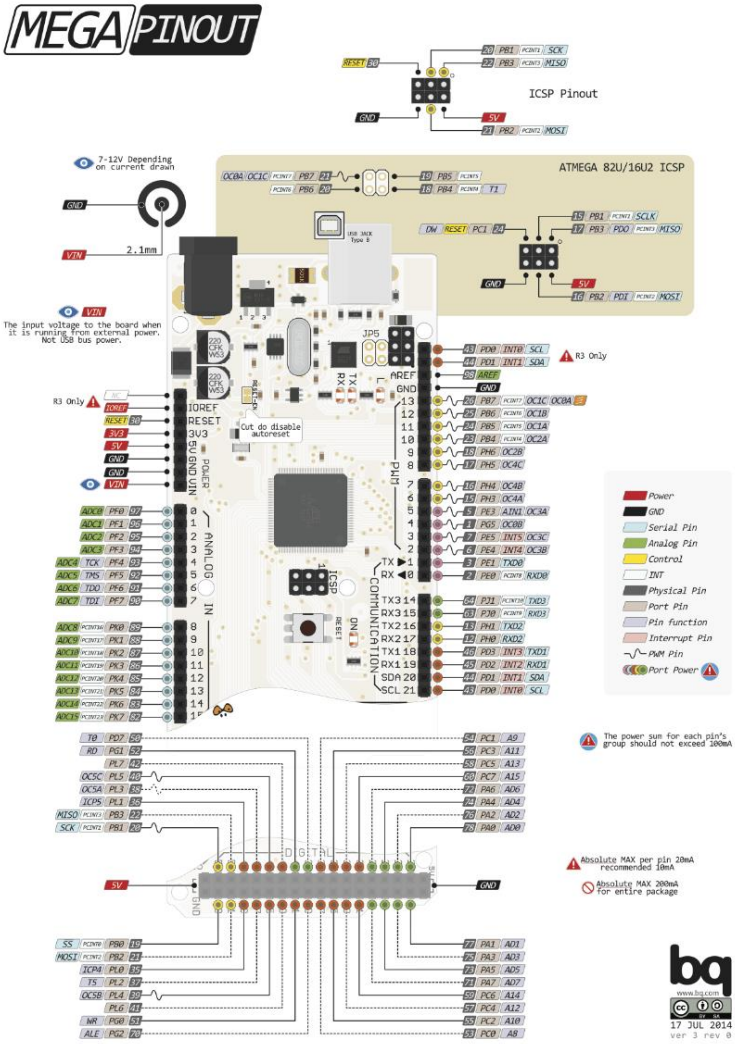
$$\rho = \rho_o * (1 + \alpha * \Delta T) \tag{2.13}$$

$$\Delta T = P * (\theta / L) \tag{2.14}$$

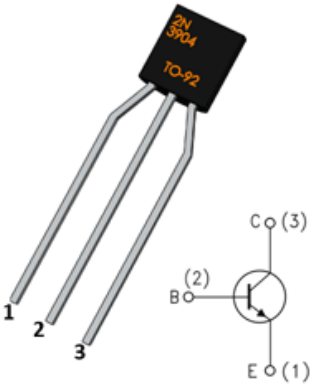
Dimana $R$ adalah resistansi kawat, $\rho$ adalah hambatan jenis kawat, L adalah Panjang kawat, $A$ adalah luas penampang penghantar kawat, $R_o$ adalah resistansi awal kawat, $\alpha$ adalah koefisien suhu resistivitas untuk jenis kawat, $\Delta T$ adalah kenaikan suhu kawat, $P$ adalah disipasi daya dalam kawat, dan $\theta$ adalah resistensi termal ke udara sekitar. Adapun bentuk fisik dari kawat nikrom dapat dilihat pada gambar 2.21.




**Gambar 2. 18.** Pin Diagram IRF540N [38]




**Gambar 2. 19.** Diagram Pin Arduino Mega [35]



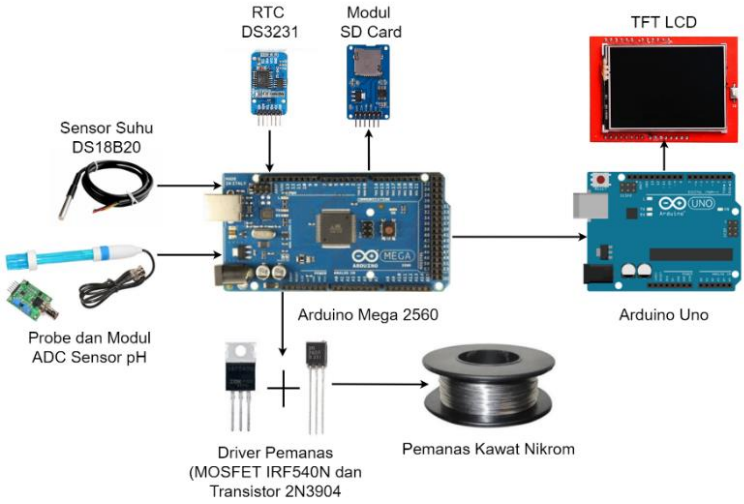

| 2N3904 | |
|---|---|
| 1 | Emitter |
| 2 | Base |
| 3 | Collector |

**Gambar 2. 20.** Pin Diagram 2N3904 [39]

**Gambar 2. 21.** Kawat Nikrom [42]

# BAB 3
# PERANCANGAN SISTEM

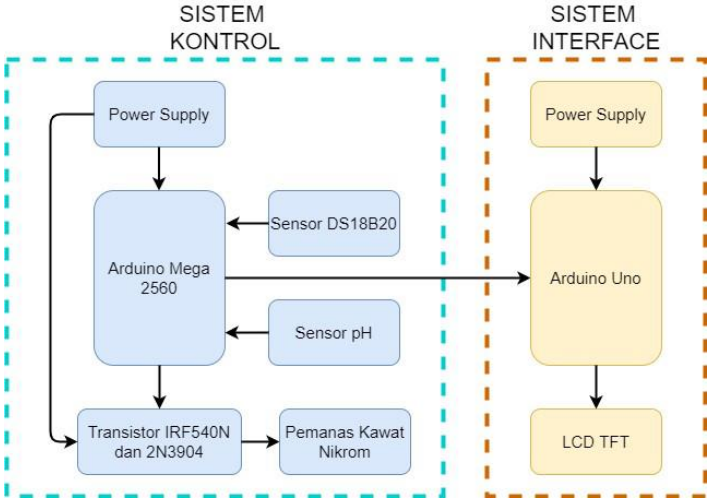
Pada bab ini dijelaskan perancangan sistem secara keseluruhan. Alat yang dirancang bertujuan untuk membuat sebuah sistem pengatur suhu pada proses fermentasi rusip dengan metode kontrol logika fuzzy serta menggunakan sensor pH dan suhu sebagai komponen umpan baliknya sehingga dapat mengontrol suhu bioreaktor atau wadah fermentasi rusip tetap optimal dan konstan sesuai dengan tingkat keasaman yang dapat mempercepat proses fermentasi serta mempermudah monitoring dalam penjagaan kualitas produk rusip. Mikrokontroler Arduino Mega akan difungsikan sebagai kontroller, komponen pembenaman metode logika fuzzy yang mengatur aktuator sesuai pembacaan data dari sensor-sensor, sedangkan Arduino Uno berperan sebagai pengolahan grafis TFT LCD selaku *interface* yang menghubungkan informasi antara manusia atau pengguna dengan perangkat. Skema keseluruhan sistem dapat dilihat pada gambar 3.1.




**Gambar 3. 1.** Skema Sistem Keseluruhan

## 3.1 Diagram Blok Sistem

Sistem ini terdiri dari dua buah subsistem yang tersusun atas Arduino Mega sebagai subsistem kontrol utama dan Arduino Uno sebagai pengolah data visualisasi parameter kontrol pada subsistem *interface*. Pada subsistem kontrol utama tersusun atas Arduino Mega, sensor suhu DS18B20, sensor pH, RTC DS3231, modul data logger SD *Card*, dan bagian aktuator pemanas kawat nikrom terdiri dari dua komponen transistor yakni MOSFET IRF540N, dan 2N3904. Sedangkan pada subsistem interface, hanya tersusun atas Arduino Uno, dan LCD TFT. Adapun diagram blok dari sistem secara keseluruhan dapat dilihat pada Gambar 3.2.




**Gambar 3. 2.** Diagram Blok Sistem Keseluruhan

- *Power Supply*
  Power supply berperan sebagai penyuplai daya yang dibutuhkan dalam sistem dihasilkan oleh adaptor 5Vdc untuk Arduino Uno dan 12Vdc untuk aktuator pemanas kawat nikrom melalui transistor IRF540N dan 2N3904.
- Arduino Uno
  Arduino Uno berperan sebagai perangkat keras yang menerima input dari sensor pH dan suhu DS18B20 yang berfungsi untuk mengendalikan tegangan aktuator pemanas berdasarkan sinyal

PWM yang diberikan melalui transistor IRF540N dan 2N3904. Selain itu Arduino Uno juga berperan sebagai komponen pemrosesan informasi parameter kontrol terhadap LCD TFT yang dikirimkan Arduino Uno pada sistem kontrol utama melalui komunikasi serial Tx/Rx.

- Sensor pH
  Sensor pH yang terdiri atas probe E201-BNC dan modul ADC pH 4502C berguna untuk mendeteksi atau umpan balik tingkat keasaman yang terkandung dalam rusip.
- Sensor Suhu DS18B20
  Berfungsi sebagai umpan balik keadaan suhu internal bioreaktor.
- RTC DS3231
  RTC DS3231 digunakan sebagai sensor untuk mendeteksi waktu atau *time stamping* guna mempermudah proses *data logging* sistem kontrol.
- LCD TFT
  LCD TFT 2,4 inchi digunakan sebagai *interface* dari sistem yang dibuat. LCD TFT menampilkan data setiap parameter yang peroleh dari hasil pembacaan sensor dan besar sinyal PWM atau duty cycle yang diberikan ke aktuator.
- Modul SD Card
  Modul SD Card digunakan sebagai komponen data logging yang merekam sekaligus mem-*backup* data-data parameter baik input maupun output yang terjadi selama sistem kontrol mengontrol proses fermentasi rusip berlangsung.
- MOSFET IRF540N
  Berperan sebagai komponen *switching* utama yang mengatur besarnya tegangan yang disalurkan menuju pemanas kawat nikrom sesuai dengan besarnya sinyal PWM yang diberikan oleh mikrokontroler Arduino Uno.
- 2N3904
  Transistor 2N3904 berfungsi sebagai *switching* pendukung antara IRF540N dan Arduino Uno, dimana 2N3904 menyesuaikan nilai output pwm Arduino Uno dari tegangan maksimum 5V. menjadi 12V sesuai dengan tegangan *threshold* pengaturan gate IRF540N.
- Kawat Nikrom
  Berfungsi sebagai aktuator yang mengubah energi listrik menjadi panas.

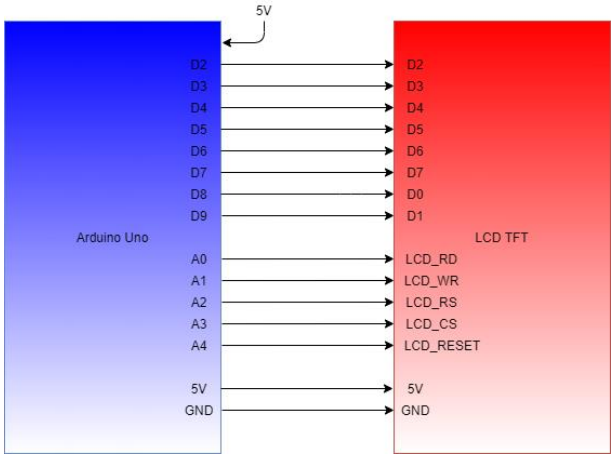
## 3.2 Perancangan Perangkat Keras

Perangkat keras (*Hardware*) yang menyusun keseluruhan sistem kontrol fermentasi rusip meliputi dua buah Arduino Uno, sensor suhu DS18B20, sensor pH, LCD TFT, *Real Time Clock* DS3231, modul data logger SD Card, kawat nikrom, transistor MOSFET IRF540N dan 2N3904.

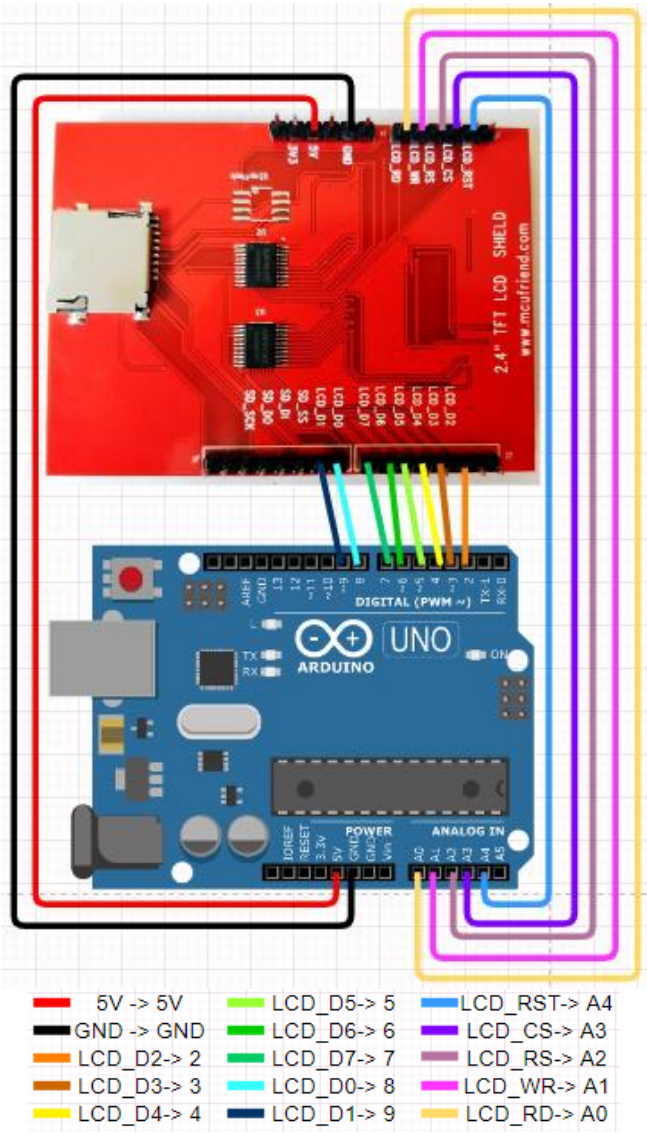
### 3.2.1 Arduino Uno dengan TFT LCD

Koneksi antara Arduino Uno dengan LCD TFT merupakan subsistem antarmuka atau *interface* yang berfungsi untuk menampilkan informasi data sensor sehingga pengguna dapat dengan mudah memantau parameter bioreaktor dan kinerja sistem kontrol suhu.

LCD TFT sendiri memiliki jumlah pin yang sama dengan pin Arduino Uno yakni 32 pin namun yang memiliki fungsi tertentu hanya 20 pin saja. komunikasi data antara Arduino Uno subsistem antarmuka dengan LCD TFT dilakukan secara parallel menggunakan pin gpio digital D2 sampai D9 untuk data dan pin analog A0-A4 digunakan untuk pengaturan instruksi, penggunaan komunikasi secara parallel dipilih dalam pengolahan grafis LCD TFT agar mengurangi waktu *delay refresh* informasi pada layar. Konektivitas antarmuka Arduino Uno dengan LCD TFT dapat dilihat pada gambar 3.3, dan 3.4 untuk skema rangkaiannya.



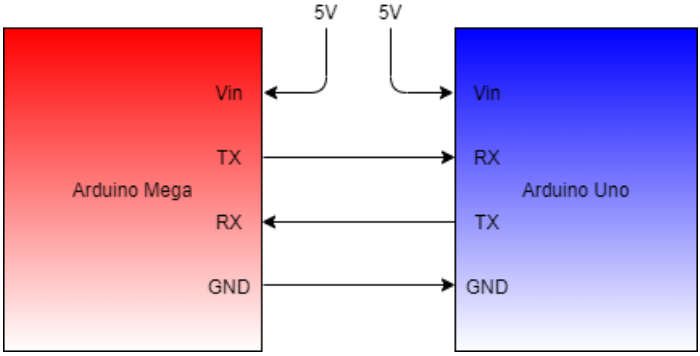

**Gambar 3. 3.** Antarmuka Arduino Uno dengan LCD TFT




**Gambar 3. 4.** Skema Rangkaian Sistem Interface antara Arduino Uno dengan LCD TFT

### 3.2.1 Antarmuka Arduino Uno dengan Arduino Mega

Dikarenakan Arduino Uno pada sistem interface hanya dioperasikan sebagai pengelola grafik akibat penggunaan pin digital hampir menyeluruh, jarak antar sistem perangkat berdekatan, serta tetap dapat memperoleh data parameter kontrol dari arduino pada sistem kontrol utama, maka dipilihlah protokol komunikasi secara serial yang akan menyalurkan data antar Arduino Uno kemudian data tersebut diproses menjadi data piksel sesuai ukuran, warna, dan posisi, Sederhananya komunikasi serial dapat menghubungkan kedua Arduino tersebut berkomunikasi satu arah seperti master dan slave via tiga buah kabel yakni Tx pin D1, Rx pin D0, dan *ground* yang saling terhubung langsung satu sama lain.

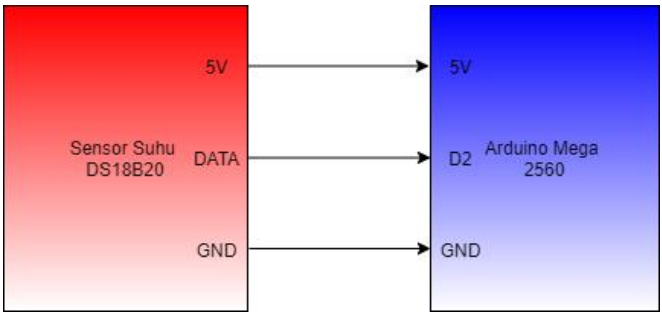



**Gambar 3. 5.** Antarmuka Arduino Mega dengan Arduino Uno

### 3.2.2 Pembacaan Sensor DS18B20

Sensor DS18B20 merupakan sensor suhu yang digunakan untuk memonitoring dan komponen umpan balik dari sistem kontrol yang bertugas mengukur parameter suhu rusip di dalam bioreaktor selama proses fermentasi berlangsung. Sensor suhu DS18B20 besutan fabrikasi Dalas Semiconductor ini telah dilengkapi library yang kompatibel digunakan pada *Integrated Development Environtment* (IDE) Arduino serta dapat diunduh secara langsung melalui Arduino IDE. Keunggulan penggunaan library ini tidak lain adalah faktor kemampuan kalibrasi pengukuran sensor yang diproses mikrokontroler Arduino tergolong sudah optimal pengaturannya sehingga mempermudah pengguna dalam hal mengatur proses pengkonversian nilai pengukuran atau nilai *Analog to Digital*

*Converter* (ADC) sensor, dimana semula berupa nilai analog diubah menjadi nilai digital, nilai hasil konversi pengukuran suhu yang disajikan dalam satuan derajat celcius. Pada intinya di bagian ini, Arduino Mega yang berada pada subsistem kontrol utama berperan sebagai perangkat yang menerima data umpan balik berupa parameter suhu, yang mana dilakukan penyesuaian bentuk data berdasarkan konteks isi pemrograman library agar tersusun data suhu yang disajikan sesuai dengan bentuk data masukan yang selanjutnya dapat diterapkan pada perhitungan kontrol logika fuzzy.

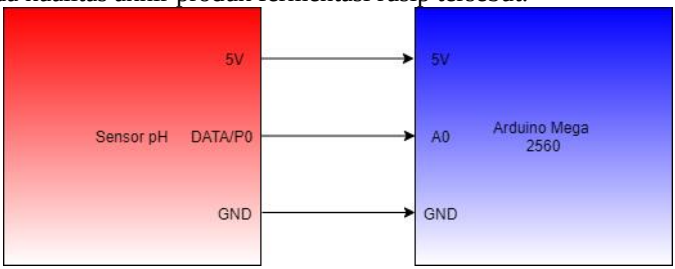



**Gambar 3. 6.** Antarmuka Arduino Uno dengan Sensor Suhu

### 3.2.3 Pembacaan Sensor PH

Berbeda cerita dengan sensor suhu DS18B20, sensor pH tidak dibekali dengan library bawaan yang sudah dapat digunakan, akan tetapi hal yang sama diantara kedua sensor adalah kontektivitas kedua sensor baik pH maupun suhu sama-sama terhubung Arduino Mega 2560 yang terletak pada subsistem kontrol utama. Sensor pH sendiri terdari atas dua bagian yakni probe pH dan modul ADC pH4502C. probe pH dapat mengeluarkan tegangan pengukuran dengan rentang pengukuran mulai dari -414,12 mV hingga 414,12 mV terhadap tingkat keasaman 0-14, sedangkan ketika diproses modul ADC dihasilkan data pengukuran berlogika 5V dengan besaran pengukuran tegangan pH netral sekitar 2,5V, pH basa kuat atau pH 14 sekitar 3,07V, dan pH asam kuat atau pH 2 sekitar 1,33V. Sensor pH digunakan sebagai umpan balik parameter kadar pH yang terkandung di dalam rusip. Data tingkat keasaman ini selanjutnya diproses oleh Arduino Mega 2560 selaku kontroller logika fuzzy bersama parameter suhu guna mengubah setpoint suhu terhadap tingkat keasaman

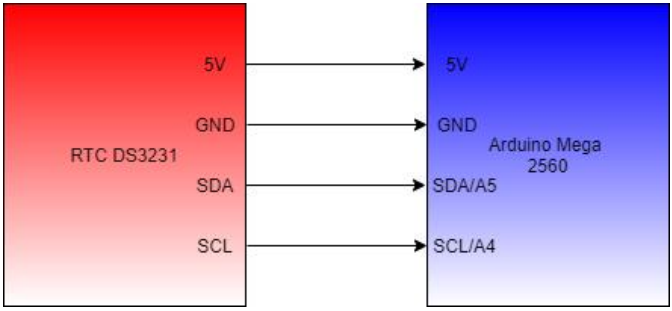
tertentu, sehingga proses fermentasi bakteri asam laktat tetap berada pada titik optimal serta memperoleh keseragaman tingkat kesamaan pada kualitas akhir produk fermentasi rusip tersebut.




**Gambar 3. 7.** Antarmuka Arduino Mega dengan Sensor PH

### 3.2.4 Arduino Mega dengan Modul RTC DS3231

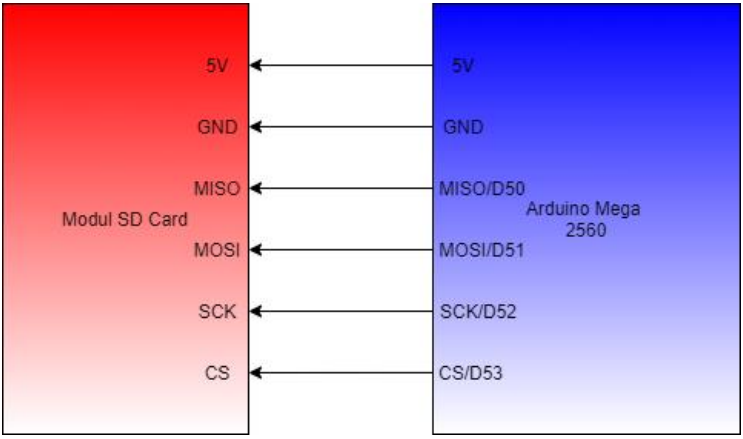
DS3231 merupakan salah satu modul *Real Time Clock* (RTC), dimana RTC adalah perangkat elektronik dikemas dalam bentuk chip yang dapat menghitung waktu (mulai dari detik hingga tahun) dengan akurat dan menjaga/menyimpan data waktu tersebut secara real time. Pada rangkaian ini, RTC digunakan untuk memberikan informasi penanggalan atau *time stamp* setiap kali pengambilan data sensor dan aktuator dilakukan. Untuk konektivitas antarmuka antara Arduino Mega dan DS3231 menggunakan protokol I2C, komunikasi data melalui port utama SDA dan SCL seperti terlihat pada gambar 3.8.




**Gambar 3. 8.** Antarmuka Arduino Mega dengan Modul RTC DS3231

### 3.2.5 Arduino Mega dengan Modul SD Card

Modul SD Card adalah modul yang memfasilitasi penyimpanan data sensor pH, suhu, dan data nilai pwm yang disalurkan ke aktuator pada kontrol sistem fermentasi rusip. penyimpanan data akan disimpan ke dalam kartu mikro SD. Kartu mikro SD dianggap memudahkan penyimpanan data karena portabilitas dan fleksibilitas pemindahan data ke berbagai perangkat elektronik seperti komputer atau *smartphone* sehingga juga memudahkan proses analisa dan evaluasi terhadap kinerja, dan kehandalan alat. Dalam hal pemindahan data sistem kontrol yang telah diolah oleh Arduino Mega menuju modul sd card dan pada akhirnya tersimpan pada mikro SD dapat terwujud melalui pengunaan protokol komunikasi *Serial Pheriperal Interface* (SPI). Komunikasi jenis ini pada intinya dilakukan melalui 4 buah port diantaranya yakni MISO pin 50, MOSI pin 51, SCK pin 52, dan CS pin 53.

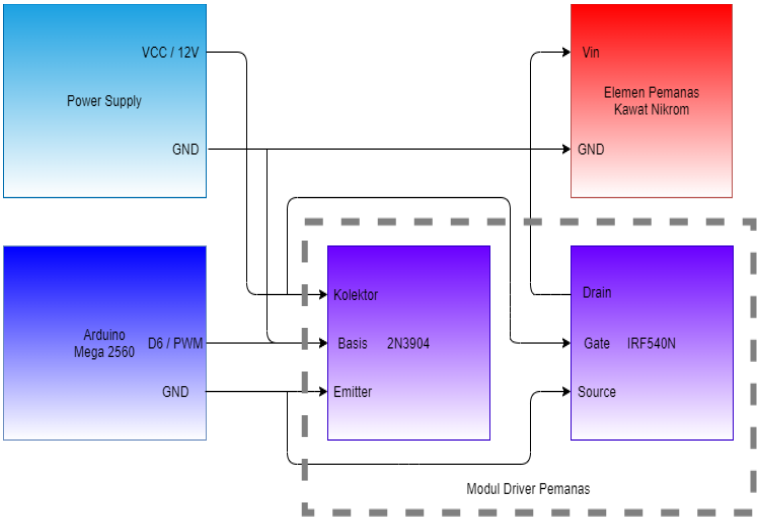



**Gambar 3. 9.** Antarmuka Arduino Mega dengan Modul Datalogging SD Card

### 3.2.6 Arduino Uno dengan Aktuator Pemanas

Arduino Mega juga digunakan sebagai pengirim perintah kepada komponen 2N3904, bagian awal dari modul driver pemanas. Secara konektivitas modul driver pemanas terhubung dengan Arduino Mega melalui pin digital D6, pada intinya, Arduino Mega pada bagian

ini berfungsi sebagai mikrokontroler utama yang mengatur modul *driver* pemanas. Driver pemanas diatur berdasarkan nilai gelombang PWM dari hasil defuzzifikasi kontrol logika fuzzy yang diterapkan pada Arduino Mega, tentunya setelah pin D6 diset menjadi pin keluaran PWM. Nilai PWM yang dihasilkan akan bervariasi, sangat bergantung terhadap besar pengukuran parameter suhu dari sensor suhu dan sensor pH yang mengatur *setpoint* suhu input awal. Adapun input dari gelombang PWM tersebut dalam modul *driver* pemanas akan melalui tahapan penyesuaian tegangan logikanya atau *buffer* dari 5V menjadi 12V oleh transistor BJT 2N3904 sebelum dialirkan ke pin *gate* MOSFET IRF540N, sehingga Arduino Mega dapat melakukan deprogram nilai *delay* yang diperlukan untuk mengatur PWM dari penyaluran arus DC *power supply* 12V/20A yang akan diteruskan ke pemanas kawat nikrom. Selain itu penggunaan MOSFET IRF540N yang terdapat pada modul *driver* pemanas dapat mengatur beban maksimal hingga 33A sehingga cocok diimplementasikan dalam pengunaan pengaturan kawat nikrom yang cenderung menggunakan arus besar untuk menghasilkan panas.




**Gambar 3. 10.** Antarmuka Arduino Mega dengan Aktuator Pemanas

### 3.2.7 Arduino Uno

Arduino Uno pada intinya berperan dalam pengolahan grafik dan visualisasi data sebelum ditampilkan pada LCD TFT. Parameter pengolahan grafik yang diatur oleh Arduino Uno adalah posisi, warna, bentuk, ataupun ukuran piksel tampilan, dimana data yang diproses merupakan data parameter kontrol baik itu parameter input seperti setpoint dan suhu ataupun parameter output hasil kontrol berupa besaran nilai *duty cycle* gelombang PWM yang diterapkan pada pemanas kawat nikrom oleh Arduino Mega 2560.

### 3.2.8 Pemanas Kawat Nikrom

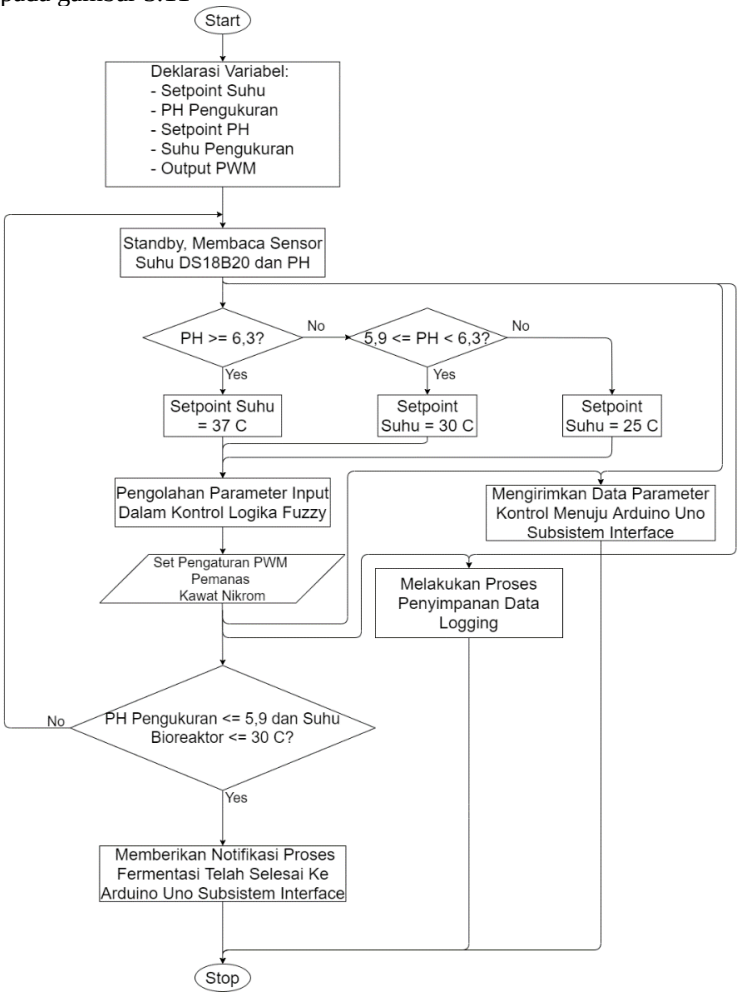
Kawat nikrom adalah elemen pemanas yang mengubah energi listrik menjadi energi panas, Pada prinsipnya kawat nikrom sama seperti kawat lainnya memiliki nilai resistansi berdasarkan jenis material penyusun kawat tersebut. Panas yang ditimbulkan kawat nikrom terjadi akibat perambatan arus yang mengalir dalam kawat dengan besaran resistansi tertentu sehingga terjadi disipasi daya atas dasar hukum Kirchoff.

Pada rangkaian ini, kawat nikrom digunakan sebagai elemen pemanas mengatur suhu internal bioreaktor berdasarkan perintah dari mikrokontroller. Jika suhu rusip dalam bioreaktor dingin, maka mikrokontroller akan membesarkan gelombang PWM ke modul *driver* agar kawat pemanas aktif, sehingga kawat nikrom menghasilkan panas, menaikkan suhu rusip, sebaliknya ketika suhu rusip panas, maka mikrokontroller akan mengecilkan gelombang PWM ke modul driver agar kawat pemanas kembali keadaan mati, sehingga suhu rusip akan menurun.

## 3.3 Perancangan Perangkat Lunak

Perancangan perangkat lunak pada tugas akhir ini meliputi pembacaan sensor suhu dan pH, penyusunan lookup table konversi data kadar pH menjadi setpoint suhu, pengiriman data dari Arduino Mega menuju Arduino Uno selaku pengolahan grafis atau tampilan dari data hasil pembacaan Arduino Mega ke dalam LCD TFT, serta penerapan sistem kontrol logika fuzzy dalam pengaturan suhu fermentasi dengan cara mengatur besar kecil gelombang PWM yang diberikan ke aktuator pemanas kawat nikrom sesuai hasil pengolahan data pengukuran sensor suhu. Adapun alur algoritma dari perancangan

perangkat lunak dari tugas akhir ini secara keseluruhan dapat dilihat pada gambar 3.11




**Gambar 3. 11.** *Flowchart* Sederhana Sistem Kontrol Keseluruhan

### 3.3.1 Pembacaan Sensor Suhu DS18B20

Pembacaan sensor suhu dipermudah karena telah tersedianya *library* khusus sensor suhu DS18B20 pada Arduino IDE. Agar *library* sensor suhu DS18B20 ini dapat memperoleh besaran satuan celcius yang dapat diproses lebih lanjut oleh mikrokontroler diawali dengan tahapan pembacaan data mentah hasil pengukuran sensor terlebih dahulu. Perlu diketahui bahwa untuk data dapat dibaca dan dimengerti oleh mikrokontroler, data hasil proses *library* masih diperlukan deklarasi penetapan pin input dari pin Arduino Mega yang akan dihubungkan ke sensor suhu DS18B20. Untuk dapat mengakses data suhu sensor DS18B20 cukup dengan menginstal *library moving average* ke dalam folder *library* Arduino serta memasukan dan mengaktivasi *library* oneWire.h dan DallasTemperature.h pada *sketch* program Arduino IDE.

```
#include <OneWire.h>
#include <DallasTemperature.h>
#include <Wire.h>

#define ONE_WIRE_BUS 2
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
float presentvalue, temporaryl;

void loop()
{
 static char temp[5];
 sensors.requestTemperatures();
 presentvalue = sensors.getTempCByIndex(0);
 if(presentvalue==-127.0)
 {
  presentvalue = temporaryl;
 }
 dtostrf(presentvalue, 4, 2, temp);
}
```

**Gambar 3. 12.** Program Pembacaan Sensor Suhu DS18B20

Penggunaan kondisi if pada program digunakan untuk memfilter dan menghindari kesalahan pembacaan sensor, dimana

ketika sensor gagal membaca akibat terjadi gangguan kabel maka sensor akan selalu mengeluarkan nilai -127.0, karena itu ditambahkan pengecekan nilai kesalahan pembacaan sensor dan menggantikannya menjadi nilai pengukuran sebelumnya.

### 3.3.2 Pembacaan Sensor PH

Prinsip kerja dari pengukuran sensor pH ini cukup mudah, dimana seperti sensor suhu DS18B20 diawali proses pembacaan data mentah. Sayangnya tidak seperti sensor DS18B20, sensor pH tidak dilengkapi dengan library bawaan yang tersedia di Arduino IDE oleh karena itu data sensor pH ini masih perlu dilakukan tahapan pengolahan seperti proses ADC menjadi bentuk 10 bit yang dapat dimengerti dan diproses lebih lanjut oleh Arduino Mega dan penerapan penstabilan nilai pengukuran dari pembacaan data mentah. Dalam hal konversi data mentah menjadi 10 bit dilakukan dengan menggunakan persamaan (3.1) dapat dilihat sebagai berikut:

$$Data\ ADC = \frac{(data\ mentah) * 5{,}0}{1024} \tag{3.1}$$

Agar mendapatkan nilai pengukuran pH dengan rentang 0-14 dilakukan penyesuaian dan kalibrasi berdasarkan karakteristik persamaan linear sensor pH. Adapun persamaan linear untuk mengkalibrasi dan menyesuaikan data sensor pH secara software dapat dilihat pada persamaan (3.2) yakni sebagai berikut:

$$Data\ ADC * (-5{,}7) + 21{,}75 \tag{3.2}$$

Dari hasil data yang diperoleh, langkah terakhir adalah memasukannya ke dalam array berukuran 60 buah data dengan *sampling* data perhitungan setiap 30 buah data untuk merata-rata besar hasil pengukuran pH per detik. Pemilihan *sampling* data setiap 30 buah data untuk dirata-rata agar didapatkan data pengukuran pH yang cenderung stabil dan memiliki persebaran data sempit atau standar deviasi kecil. Perhitungan rata-rata data dari array secara per sampling dapat diperoleh dengan menggunakan library *Moving* Average. Sama seperti penjelasan dibagian sebelumnya, untuk mengakses *library moving average* ini cukup dengan menginstal *library MovingAverage.h* ke dalam *folder library* Arduino serta memasukan dan mengaktivasi pada *sketch* program Arduino IDE.

```
#include "Arduino.h"
#include "MovingAverage.h"

const int analogInPin = A0;
float nilaipH, tableTest[60];
// m = (pH7 - pH4) / (Vph7 - Vph4)
const float m = -7.059;
MovingAverage <float> test(30);

void loop()
{
  float Po = analogRead(analogInPin) * 5.0 / 1024;
  float phValue = 7 - (2.725 - Po) * m;
  for(int i = 0; i<60; i++)
  {
   tableTest[i] = {phValue};
   nilaipH = test.CalculateMovingAverage(tableTest[i])
   delay(10);
  }
}
```

**Gambar 3. 13.** Program Pembacaan Sensor pH

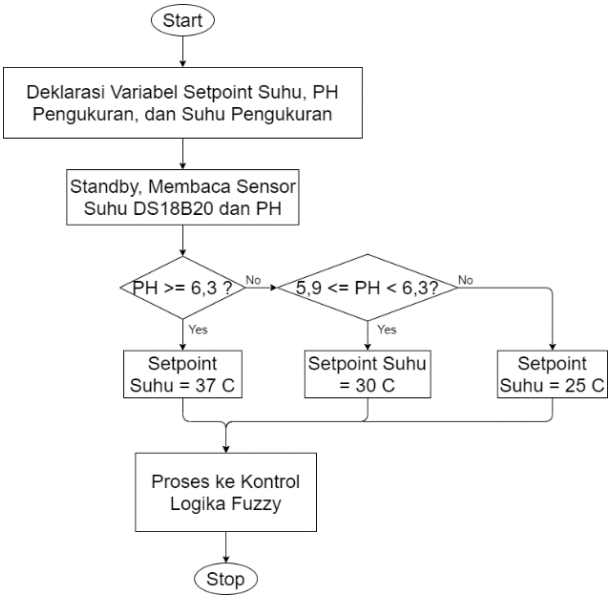
### 3.3.3 Penyusunan Lookup Table Konversi PH Menjadi Setpoint Suhu

Fungsi dari *lookup table* ini sebetulnya cukup sederhana yakni mengonversikan atau mengubah hasil pembacaan pengukuran tingkat keasaman menjadi setpoint suhu yang sesuai berdasarkan pedoman rentang optimum dari tingkat keasaman terhadap suhu optimum dari masing-masing bakteri asam laktat yang terlibat disetiap tahapan proses fermentasi rusip. Adapun pertimbangan rentang nilai tingkat keasaman atau pH yang dimasukan berjumlah 4 buah karena bakteri asam laktat yang terlibat dalam proses fermentasi ada 3 jenis bakteri asam laktat sehingga terdapat 4 buah tahapan atau rentang *lookup table*, pada tahap awal yaitu bakteri *Streptococcus* dengan rentang tingkat keasaman kisaran 6 hingga 7 atau tepatnya lebih dari 6,2 dan suhu optimum 37ºC, tahapan kedua yaitu bakteri *Lactococcus* dengan rentang pH 6,9 hingga 6,3, dan suhu optimum 30 ºC, tahapan fermentasi yang terakhir dilakukan oleh bakteri *Leuconostoc* dengan

rentang pH 6,5 hingga 5,5, dan suhu optimum berkisar antara 30 ºC hingga 25 ºC. Fungsi utama dari *lookup table* ini ialah pengubahan data umpan balik sensor pH menjadi parameter suhu yang akan digunakan sebagai input *setpoint* dari kontrol logika fuzzy, Adapun bentuk penyusunan desain tabel *lookup table* sendiri dapat dilihat pada tabel 3.1 dan gambaran detail *flowchart* dari pengubahan parameter pH menjadi suhu setpoint oleh *lookup table* dapat dilihat pada gambar 3.14.

**Tabel 3. 1.** *Lookup Table* Konversi PH Menjadi Setpoint Suhu

| Rentang pH | Setpoint Suhu (ºC) |
|---|---|
| >= 6,3 | 37 |
| 6,3 – 5,9 | 30 |
| <= 5,9 | 25 |




**Gambar 3. 14.** *Flowchart* Lookup Table Mengonversikan Parameter PH Menjadi Parameter Suhu Setpoint

### 3.3.4 Pengiriman Data antara Arduino Mega dengan Arduino Uno

Pengiriman data dari Arduino Mega menuju Arduino Uno dapat dilaksanakan setelah seluruh parameter-parameter sistem kontrol telah diperoleh, baik itu parameter input maupun output. Dimana nantinya data-data tersebut pada akhirnya akan ditampilkan pada LCD TFT. Pengiriman data dilakukan secara bersamaan, yang mana akan disusun ke dalam satu *array* panjang, sehingga meningkatkan efisiensi waktu serta menghindari kesalahan data pengiriman yang kedepannya berpeluang mempengaruhi pengolahan grafis atau visualisasi data oleh Arduino Uno pada subsistem *interface*. Untuk detail dari algoritma pemograman pengiriman data antara kedua Arduino pada bagian sisi Arduino Mega 2560 dalam subsistem kontrol dapat dilihat pada gambar 3.15 sedangkan penerimaan data dapat dilihat pada gambar 3.16.

```
1  float nilaipH, presentvalue, setpoint, outputPWM;
2
3  void loop()
4  {
5   static char temp[5];
6   static char pH[5];
7   static char set[5];
8   static char pwm[5];
9   dtostrf(presentvalue, 4, 2, temp);
10  dtostrf(nilaipH, 3, 2, pH);
11  dtostrf(setpoint, 4, 2, set);
12  dtostrf(outputPWM, 5, 2, pwm);
13  char buff[25];
14  sprintf(buff,"<%.5s %.5s %.5s %.5s>\n", temp, pH, set, pwm)
15  Serial.write(buff);
16 }
```

**Gambar 3. 15.** Program Pengiriman Data Arduino Mega Subsistem Kontrol

Penggunaan fungsi sprintf adalah untuk melakukan pengeluaran nilai parameter kontrol baik input seperti suhu, tingkat keasaman, dan setpoint maupun komponen parameter output nilai PWM secara bersamaan disalurkan dalam komunikasi serial.

```
const byte numChars = 25;
char receivedChars[numChars];
char tempChars[numChars];
float datemp = 0.0;
float nilaipH = 0.0;
float setpoint = 0.0;
float outputPWM = 0.0;
boolean newData = false;

void loop
{
recvWithStartEndMarkers();
    if (newData == true)
    {
    strcpy(tempChars, receivedChars);
    parseData();
    newData = false;
    }
}
```

**Gambar 3. 16.** Program Penerima Data Kontrol Arduino Uno Subsistem Interface

Agar Arduino Uno subsistem *interface* mampu mengenali awal dan akhir data setiap pengiriman maka diperlukan pemberian penandaan awal data dan akhir data seperti yang ditunjukkan pada gambar 3.17. dan diperlukan *parsing* atau pemisahan data kembali masing-masing data sesuai *jenis* parameternya yang dapat diamati pada gambar 3.18.

```
void recvWithStartEndMarkers()
{
    static boolean recvInProgress = false;
    static byte ndx = 0;
    char startMarker = '<';
    char endMarker = '>';
    char rc;
    while (Serial.available() > 0 && newData == false){
        rc = Serial.read();
        if (recvInProgress == true) {
            if (rc != endMarker) {
                receivedChars[ndx] = rc;
                ndx++;
                if (ndx >= numChars) {
                    ndx = numChars - 1;
                }
            }
            else {
                receivedChars[ndx] = '\0';
                recvInProgress = false;
                ndx = 0;
                newData = true;
            }
        }
        else if (rc == startMarker) {
            recvInProgress = true;
        }
    }
}
```

**Gambar 3. 17.** Program Pembaca Penanda Awal dan Akhir Data

```
void parseData()
{
    char * strtokIndx;
    strtokIndx = strtok(tempChars," ");
    datemp = atof(strtokIndx);
    strtokIndx = strtok(NULL, " ");
    nilaipH = atof(strtokIndx);
    strtokIndx = strtok(NULL, " ");
    setpoint = atof(strtokIndx);
    strtokIndx = strtok(NULL, " ");
    outputPWM = atof(strtokIndx);
}
```

**Gambar 3. 18.** Program Pemisah Data Parameter Kontrol

### 3.3.5 Tampilan Interface Parameter Kontrol Pada LCD TFT

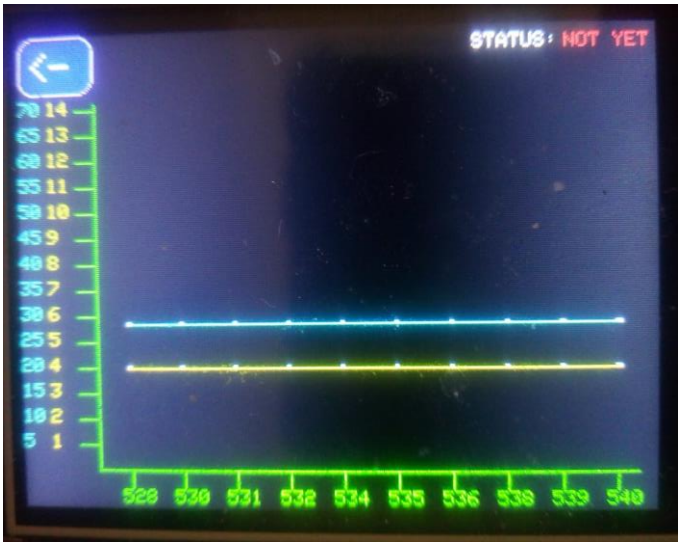
Sesudah data sensor dan aktuator diterima Arduino Uno pada subsistem *interface*, guna memperjelas kinerja sistem kontrol, dan memudahkan penjelasan terhadap apa yang terjadi secara visual, maka data-data tersebut perlu diolah kembali bentuk dan pengaturan visualisasinya oleh Arduino Uno sebelum ditampilkan ke LCD TFT.

Jumlah tampilan antarmuka terbagi atas empat buah *slide* seperti yang terlihat pada gambar 3.19, 3.20, 3.21, dan 3.22, dimana keempat *slide* layar tersebut antara lain adalah tampilan menu utama antarmuka, tampilan antarmuka submenu temperatur, tampilan antarmuka submenu sensor pH, dan tampilan submenu grafik historis yang memetakan data temperature data pH. Selain itu modul LCD TFT telah dilengkapi dengan fitur *touchscreen*, *sehingga* membantu pengguna bernavigasi antar layar menu dan submenu dari 4 *slide* tersebut. Berikut ini merupakan contoh tampilan antarmuka menu dan submenu dari empat *slide* yang telah dijelaskan sebelumnya, dimana berfungsi untuk mengatur bentuk visualisasi data atau parameter sistem kontrol pada LCD TFT.

**Gambar 3. 19.** Tampilan Menu Utama Antarmuka

**Gambar 3. 20.** Tampilan Antarmuka Submenu Temperatur

**Gambar 3. 21.** Tampilan Antarmuka Submenu pH




**Gambar 3. 22.** Tampilan Antarmuka Submenu Grafik Historis

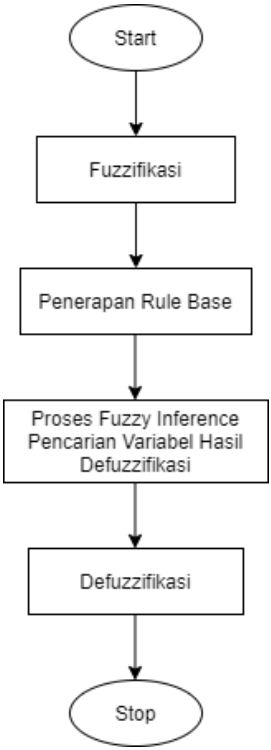
### 3.3.6 Sistem Kontrol Logika Fuzzy

Pada penelitian ini, metode kontrol yang digunakan adalah metode kontrol logika fuzzy. Logika fuzzy dipilih diimplementasikan pada sistem kontrol ini karena karakteristik dari metode logika fuzzy dianggap mudah untuk dimengerti akibat penggunaan pembahasan kontrol melalui pendekatan *human linguistic* atau bahasa manusia, tidak hanya itu, metode logika fuzzy juga menawarkan kemampuan kontrol tertentu yang tidak dimiliki oleh metode lainnya seperti kehandalan, fleksibilitas pengimplementasian dan *robust* atau ketahanan sistem kontrol menjaga keadaan parameter kontrol dari adanya gangguan eksternal diluar dari *plant* kontrol, Pada tugas akhir ini, input sistem yang digunakan terdiri dari dua buah sensor, sensor suhu dan pH. Besar hasil pengukuran sensor suhu akan mempengaruhi *duty cycle* pemanas yang ditentukan dari besarnya pemberian gelombang PWM keluaran mikrokontroler menuju modul *driver* pemanas yaitu MOSFET IRF540N dan transistor 2N3904. Sedangkan pembacaan sensor pH akan mempengaruhi titik setpoint suhu, dimana setpoint tersebut besar nilainya akan bervariasi terhadap rentang tingkat keasaman tertentu yang mewakili keadaan optimum

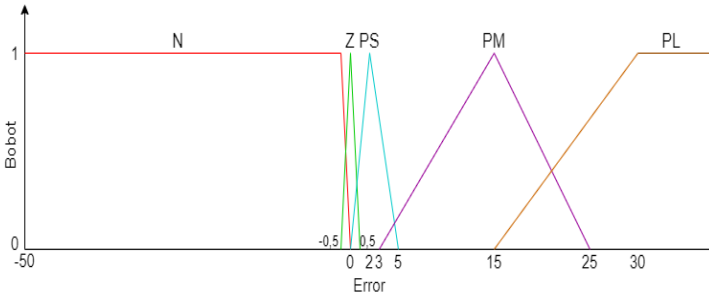
pertumbuhan bakteri asam laktat yang terlibat pada proses fermentasi. Tahap pertama dari perancangan sistem kontrol logika fuzzy ialah fuzzifikasi, dimana tahap fuzzifikasi merupakan tahapan penentuan bentuk fungsi keanggotaan yang berperan mengubah parameter input atau crisp input menjadi himpunan fungsi keanggotaan. Pada percobaan alat hanya diperlukan satu setpoint yakni suhu, namun dalam perjalanan proses fermentasi setpoint tersebut akan berubah-ubah sesuai isi desain *lookup table* pengkonversi parameter tingkat keasaman yang diukur sensor pH menjadi suhu sebelum diproses ke dalam kontrol logika fuzzy.

Input pada kontrol logika fuzzy ini adalah error dan Δerror, dimana error merupakan setpoint dikurangi pengukuran sekarang. Sedangkan Δerror merupakan hasil error sekarang dikurangi error sebelumnya. Karena itu, pada tahap fuzzifikasi di awal tahapan ini melakukan penentuan fungsi keanggotaan input error dan Δerror terlebih dahulu. Fungsi keanggotaan error adalah *negative* ketika suhu pengukuran sensor dibawah suhu setpoint, *zero* ketika suhu pengukuran sensor tepat di titik setpoint, *positif small* ketika suhu pengukuran sensor sedikit melebihi setpoint, *positif medium* ketika suhu pengukuran sensor melebihi setpoint, dan *positif large* ketika suhu pengukuran sensor jauh melebihi setpoint. Sedangkan pada fungsi keanggotaan Δerror adalah *negative large* ketika selisih error sangat bernilai negatif, *negative medium* ketika selisih error bernilai negatif, *negative small* ketika selisih error sedikit bernilai negatif, *zero* ketika selisih error tidak ada, *positive small* ketika selisih error sedikit bernilai sedikit positif, *positive medium* ketika selisih error bernilai lebil positif, dan terakhir *positive large* ketika selisih error jauh lebih positif atau lebih besar.

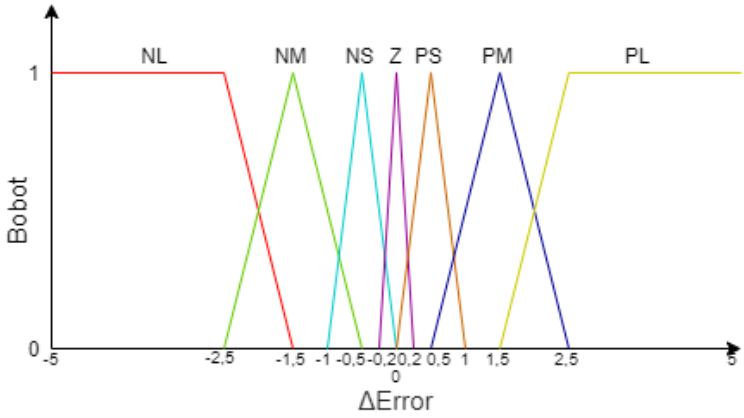
Adapun *flowchart* sistem logika fuzzy dapat diamati pada gambar 3.23. sedangkan fungsi keanggotaan dari error dan Δerror sistem fuzzy yang telah dirancang masing-masing dapat dilihat pada gambar 3.24 dan 3.25.




**Gambar 3. 23.** *Flowchart* Sistem Logika Fuzzy




**Gambar 3. 24.** Fungsi Keanggotaan Input Error




**Gambar 3. 25.** Fungsi Keanggotaan Input Δerror

Layaknya sebuah sistem kontrol, fungsi keanggotaan juga digunakan dalam menentukan output dari hasil proses pengolahan input pada bagian inferensi fuzzy, dimana fungsi keanggotaan di tugas akhir ini mengatur parameter keluaran output gelombang PWM pemanas. Berikut ini merupakan gambar fungsi keanggotaan output nilai gelombang PWM yang akan disalurkan ke pemanas kawat nikelin dan gambar fungsi keanggotaan output setpoint suhu dari pengolahan parameter pH, secara detailnya masing-masing dapat diamati pada gambar 3.24.

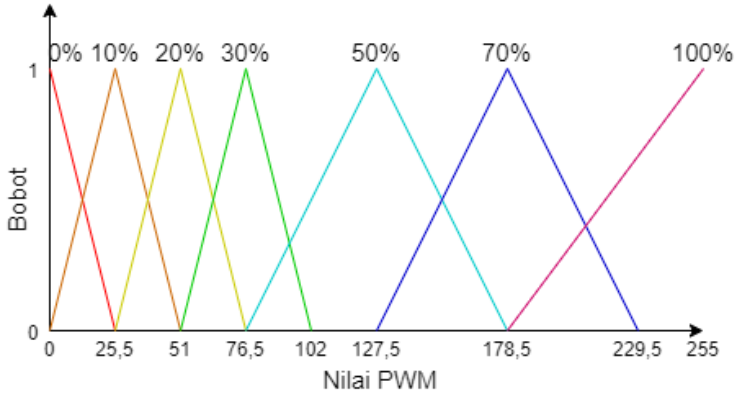
Tentunya fungsi keanggotaan input tidak dapat diproses ataupun memperoleh hasil yang sesuai pada fungsi keanggotaan output secara begitu saja. Oleh karenanya, diperlukanlah rambu-rambu panduan dasar, dimana rambu-rambu tersebut disusun mengikuti panduan dari pendapat ahli dibidangnya yang diwakilkan pada tahapan *rule base* atau aturan dasar. Adapun *rule base* terdapat pada tugas akhir ini terbagi atas dua *rule base*, yaitu *rule base* pengaturan PWM pemanas dan pengembangan fitur konversi parameter pH menjadi setpoint suhu seperti yang telah dijelaskan sebelumnya, keduanya dapat dilihat pada tabel (3.2).

**Tabel 3. 2.** Rule Base Pengaturan PWM Pemanas

| error / Δerror | Negative (N) | Zero (Z) | Positive Small (PS) | Positive Medium (PM) | Positive Large (PL) |
|---|---|---|---|---|---|
| Negative Large(NL) | 0% | 20% | 100% | 100% | 100% |
| Negative Medium (NM) | 0% | 10% | 70% | 100% | 100% |
| Negative Small(NS) | 0% | 0% | 50% | 100% | 100% |
| Zero(Z) | 0% | 0% | 30% | 100% | 100% |
| Positive Small(PS) | 0% | 0% | 20% | 70% | 100% |
| Positive Medium (PM) | 0% | 0% | 10% | 50% | 100% |
| Positive Large(PL) | 0% | 0% | 0% | 30% | 100% |

Dari tabel 3.2. dapat diamati bahwa pertama, pemanasan pemanas menggunakan gelombang PWM *duty cycle* 100% jika error secara keseluruhan pada kondisi *positive* large, error pada kondisi *positive medium* dengan Δerror mulai dari kondisi *negative large* sampai *zero*, serta error hanya pada kondisi *positive small* dengan Δerror *negative large*. Kedua, pemanasan pemanas menggunakan gelombang PWM *duty cycle* 70% diberikan jika error pada kondisi *positive medium* dengan Δerror *positive small*, dan error pada kondisi *positive small* dengan Δerror *negative medium.* Ketiga, pemanasan pemanas menggunakan gelombang PWM *duty cycle* 50% jika error dan Δerror pada kondisi *positive medium*, dan error pada kondisi *positive small* dengan Δerror *negative small*. Keempat, pemanasan pemanas menggunakan gelombang PWM *duty cycle* 30% jika error pada kondisi *positive medium* dengan Δerror *positive large*, dan error pada kondisi *positive small* dengan Δerror *zero.* Kelima, pemanasan pemanas menggunakan gelombang PWM *duty cycle* 20% jika error dan Δerror keduanya pada kondisi *positive small*, dan error pada kondisi *zero* dengan Δerror *negative large*. Keenam, pemanasan pemanas menggunakan gelombang PWM *duty cycle* 10% jika error pada kondisi *positive small* dengan Δerror *negative medium*, dan error pada kondisi *zero* dengan Δerror *negative medium*. Terakhir ketujuh,

pemanasan pemanas menggunakan gelombang PWM *duty cycle* 0% alias mati jika error secara keseluruhan *negative*, error pada kondisi *zero* dengan Δerror mulai dari kondisi *negative small* sampai *positive large*, serta error hanya pada kondisi *positive small* dengan Δerror *positive large*.




**Gambar 3. 26.** Fungsi Keanggotaan Output Nilai PWM Pemanas

Setelah menentukan aturan dasar, maka masuk ke tahap selanjutnya yaitu defuzzyfikasi, Karena sistem kontrol pada penysunan tugas akhir ini menggunakan aturan Mamdani maka pengambilan keputusan hasil defuzzifikasi diperoleh dengan metode *Center Of Gravity* (COG) atau seringkali juga dikenal dengan *Center Of Area* (COA). Metode *Center Of Gravity* ini dapat dilakukan dengan cara mengambil titik pusat daerah fuzzy, secara umum dirumuskan menjadi dua bentuk persamaan yaitu pada persamaan (3.1) untuk pengolahan variabel kontinyu dan persamaan (3.2) untuk pengolahan variabel diskrit.

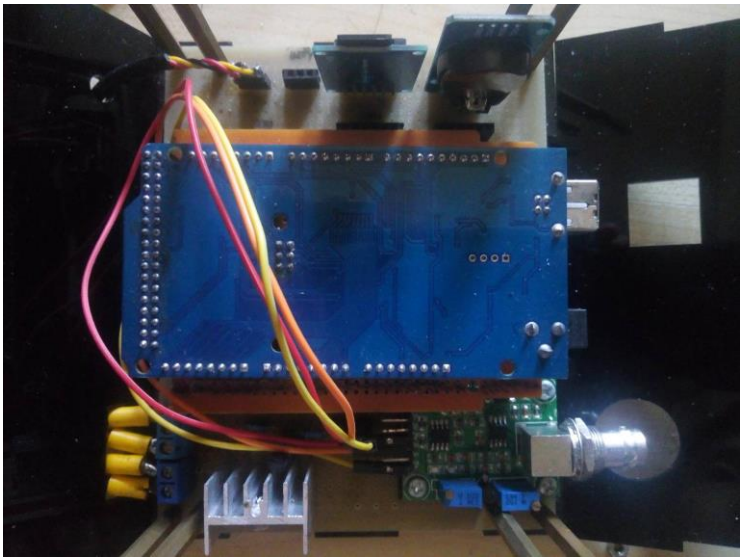
$$x_0 = \frac{\int_a^b x\mu_A(x)dx}{\int_a^b \mu_A(x)dx} \tag{3.1}$$

$$x_0 = \frac{\sum_{x=a}^{b} \mu_A(x)x}{\sum_{x=a}^{b} \mu_A(x)x} \tag{3.2}$$

*Halaman ini sengaja dikosongkan*

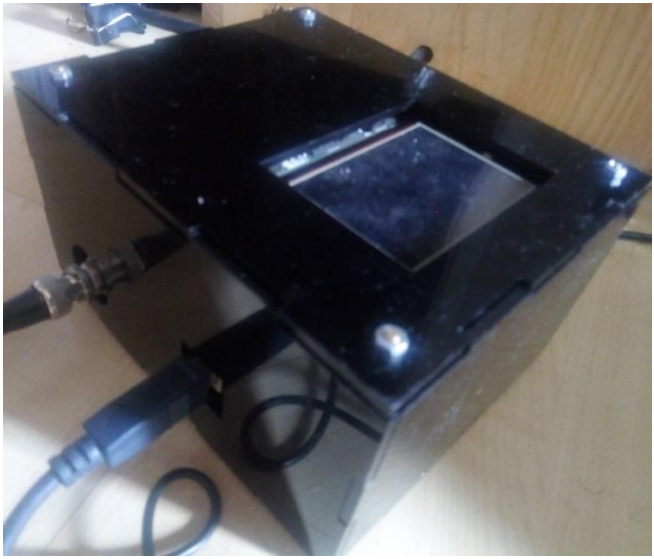
# BAB 4
# PENGUJIAN DAN ANALISIS

Bab ini akan membahas tentang pengujian hardware dan software dari sistem kontrol yang mana telah dirancang sesuai pada pembahasan bab sebelumnya, serta pada bab ini disusun untuk memperoleh analisis data dari setiap pengujian yang telah dilakukan. Pengujian dilakukan baik itu per bagian komponen penyusun alat maupun keseluruhan alat sistem kontrol suhu pada fermentasi rusip menggunakan metode logika fuzzy berdasarkan tingkat keasaman. Detail pengujian yang akan dilakukan antara lain meliputi pengujian sistem sensor, sistem kontrol logika fuzzy, sistem aktuator pemanas kawat nikrom, tampilan data parameter kontrol, serta penyimpanan *data logging*. Semua rangkaian pengujian tersebut dilakukan bertempat lokasi di dalam apartemen Puncak Kertajaya, unit A2072. Adapun gambar hasil realisasi perangkat keras sistem kontrol secara keseluruhan dapat diamati pada gambar 4.1 sampai 4.6.



**Gambar 4. 1.** Realisasi Subsistem Kontrol Utama (Dalam)

**Gambar 4. 2.** Realisasi Subsistem Interface (Dalam)



**Gambar 4. 3.** Realisasi Box Sistem Kontrol (Luar)

(a) (b)

**Gambar 4. 4.** Realisasi Pemanas Kawat Nikrom Bagian Dalam (a) dan Realisasi Pemanas Kawat Nikrom Bagian Luar (b)

**Gambar 4. 5.** Realisasi Bioreaktor Fermentasi Rusip

## 4.1 Kalibrasi Sistem Sensor

Agar memperoleh hasil pengukuran sensor yang akurat maka dilakukan tahapan kalibrasi sensor melalui pengukuran suhu dan pH

dalam jumlah data banyak dengan berbagai nilai. Kalibrasi pada tugas akhir ini dilakukan terhadap sensor suhu dan pH dengan cara menguji dan membandingkan data terhadap instrumen yang sudah presisi ataupun suatu zat yang telah memiliki nilai yang konstan.

### 4.1.1 Kalibrasi Sensor Suhu dengan Termometer

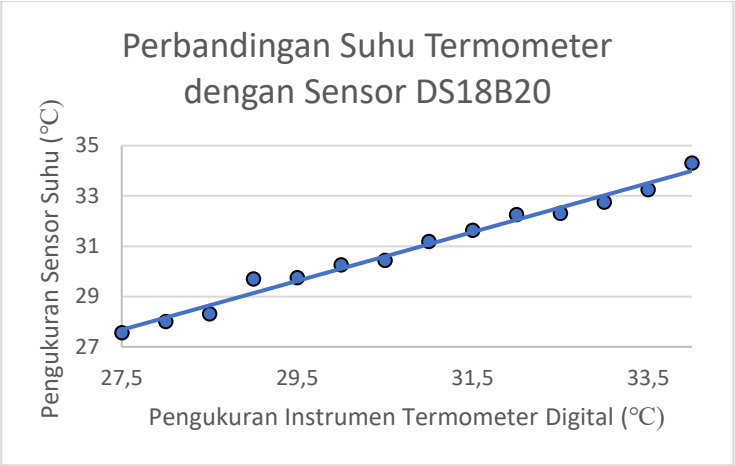
Pada kalibrasi sensor suhu DS18B20 terlebih dahulu dimasukkan ke dalam library DallasSemicondakan dilakukan dengan cara membandingkan nilai pengukuran sensor terhadap digital termometer. Setelah didapatkan hasil pengukuran maka kedua nilai sensor dan instrumen akan dihubungkan dicari tingkat kesesuaian antara sensor suhu terhadap digital thermometer, dimana pada akhirnya dapat menentukan tingkat keakuratan dan mengevaluasi pembacaan sensor suhu DS18B20. Jenis digital thermometer yang digunakan sebagai pembanding dapat diamati pada gambar 4.6 sebagai berikut:

**Gambar 4. 6.** Instrumen Termometer Digital

Kalibrasi dilakukan dengan menggunakan 14 sampel data di dalam ruangan tertutup yang dilengkapi dengan pendingin udara. Sampel suhu yang diambil untuk pengujian berkisar antara rentang 27,5℃ hingga 34℃ dengan skala kenaikan data per 0,5℃. Tujuan dari pengambilan sampel suhu dengan nominal yang berbeda-beda adalah agar mengetahui linearitas dan keakuratan pengukuran antara sensor suhu terhadap instrumen thermometer digital. Hasil kalibrasi tersebut akan digunakan untuk mengolah data pengukuran suhu dalam sistem kontrol kedepannya. Adapun hasil kalibrasi sensor terhadap digital termometer yang diperoleh dari pengujian akan disusun dalam bentuk tabel dan grafik, dimana masing-masing dapat diamati pada tabel 4.1 dan gambar 4.7.

**Tabel 4. 1.** Hasil Kalibrasi Sensor Suhu Terhadap Termometer

| Termometer (℃) | Sensor DS18B20 (℃) | Error (%) |
|---|---|---|
| 27,5 | 27,56 | 0,22 |
| 28 | 28 | 0 |
| 28,5 | 28,31 | 0,67 |
| 29 | 29,69 | 2,38 |
| 29,5 | 29,75 | 0,85 |
| 30 | 30,25 | 0,83 |
| 30,5 | 30,44 | 0,2 |
| 31 | 31,19 | 0,61 |
| 31,5 | 31,63 | 0,41 |
| 32 | 32,25 | 0,78 |
| 32,5 | 32,31 | 0,59 |
| 33 | 32,75 | 0,76 |
| 33,5 | 33,25 | 0,75 |
| 34 | 34,31 | 0,91 |



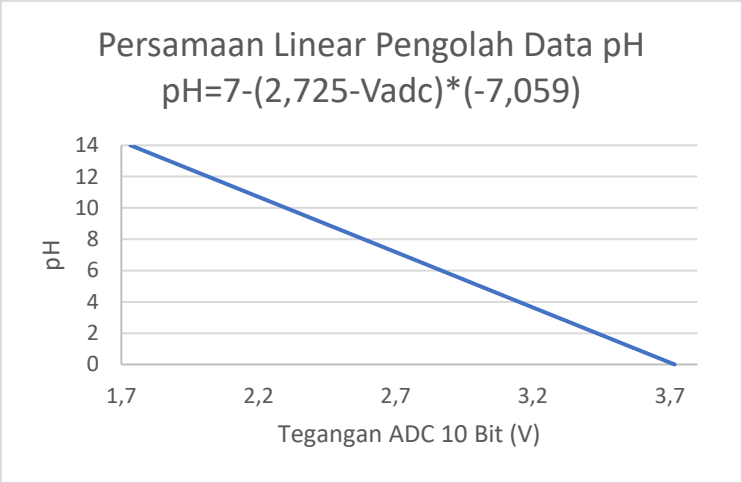

**Gambar 4. 7.** Grafik Perbandingan Suhu Termometer dengan Sensor Suhu

### 4.1.2 Kalibrasi Sensor PH dengan Cairan PH Buffer

Pada kalibrasi juga dilakukan terhadap sensor pH dengan cara membandingkan hasil pengukuran terhadap nilai asli dari cairan buffer yang dicelupkan sensor pH. Cairan buffer yang digunakan dalam kalibrasi sensor pH adalah pH 4, 6.86, 7 dan 10. Untuk adalah gambar dari ketiga cairan pH buffer dapat diamati pada gambar 4.8.

**Gambar 4. 8.** Cairan PH Buffer Solution

proses kalibrasi sebetulnya dilakukan guna menemukan persamaan linear pengakuisisi data tingkat keasaman menjadi nilai tegangan listrik yang sesuai, serta menjamin kinerja sensor pH mengukur berbagai tingkat keasaman seperti asam, netral, dan basa mampu beroperasi dengan normal. Persamaan linear akuisisi data dari sensor pH yakni $pH = 7 - (2{,}725 - V_{adc})(-7{,}059)$ nilai -7,059 sendiri adalah nilai gradien (m) dimana persamaan tersebut dapat diamati pada gambar 4.9.

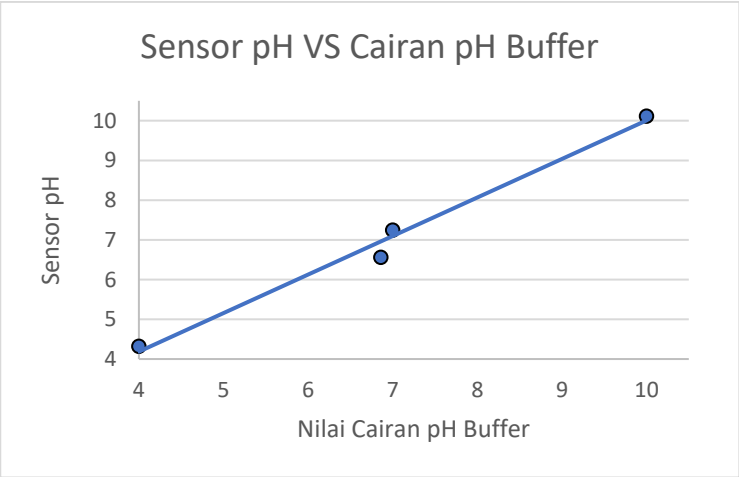



**Gambar 4. 9.** Hasil Persamaan Linear Kalibrasi Sensor pH

Perlu diketahui juga proses kalibrasi sensor pH dilakukan dengan menggunakan 3 buah sampel tingkat keasaman yang berbeda-beda, dimana diperoleh dari pengunaan cairan pH buffer 4, 6.86, dan 10 dengan banyak jumlah data per sampel sebanyak 20 dengan waktu pengambilan setiap data dalam sampling waktu 10 detik. Selain itu kalibrasi sensor juga tidak mempertimbangkan faktor parameter suhu sebagai parameter eksternal yang berpeluang mempengaruhi keakuratan pengukuran sensor pH selain kemampuan probe pH. Untuk tingkat keakuratan pengukuran dari proses kalibrasi sensor pH terhadap cairan pH buffer 4, 6.86, 7 dan 10, dapat dilihat pada tabel 4.2, sedangkan perbandingan hubungan linearitas antara sensor pH dan cairan pH buffer dapat diamati pada gambar grafik 4.10.

**Tabel 4. 2.** Hasil Pengukuran Sensor PH Terhadap Cairan PH Buffer

| Cairan pH Buffer | PH Terukur Sensor | Error |
|---|---|---|
| 4 | 4,32 | 8% |
| 6,86 | 6,56 | 4,37% |
| 7 | 7,25 | 3,57% |
| 10 | 10,12 | 1,2% |




**Gambar 4. 10.** Grafik Perbandingan Pengukuran Sensor PH dengan Cairan PH Buffer

## 4.2 Pengujian Sistem Aktuator Pemanas

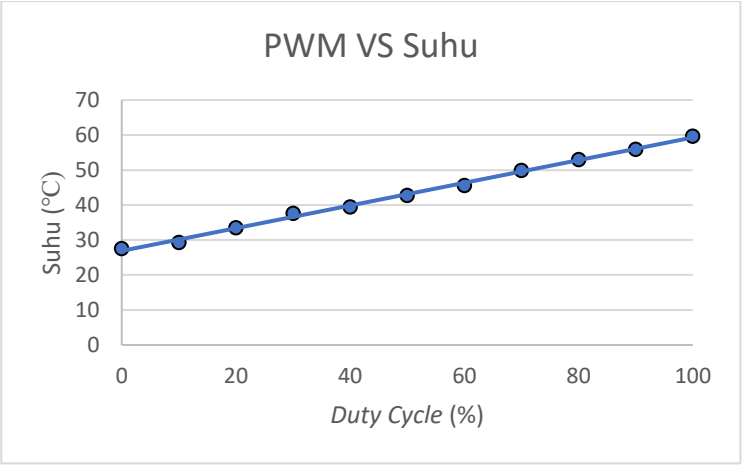
Pada pengujian ini dilakukan untuk mengetahui pengaruh pemberian nilai gelombang pwm berubah-ubah terhadap suhu yang dihasilkan oleh kawat nikelin selaku aktuator pengontrol suhu internal bioreaktor fermentasi, dimana dari hasil pengujian ini akan digunakan sebagai pedoman pengaturan kontrol logika fuzzy yang diterapkan dalam sistem kontrol. Pengujian ini dilakukan dengan temperatur ruang saat itu sebesar ±25ºC dan rentang nilai gelombang PWM mulai dari 0-255 dengan skala data per 10% *duty cycle*. Adapun hasil

pengujian pengaruh besar nilai gelombang pwm terhadap suhu yang dihasilkan dapat diamati pada tabel 4.3.

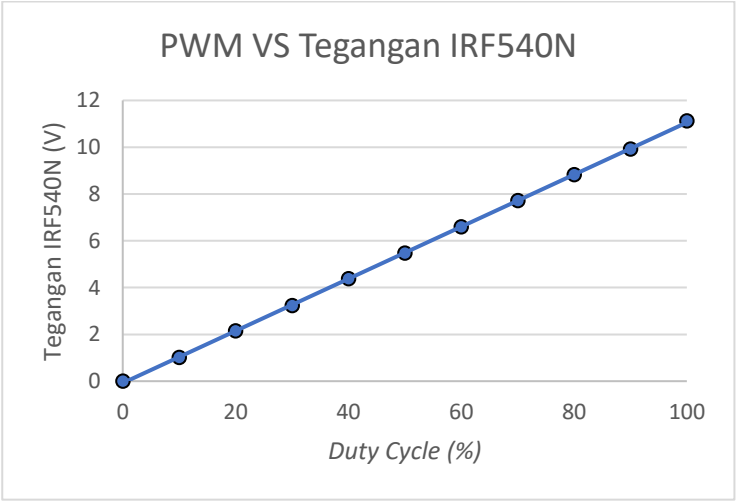
**Tabel 4. 3.** Pengaruh Nilai PWM Pemanas Terhadap Suhu

| Pin Output PWM Arduino Mega 2560 | | Driver Pemanas (IRF540N + 2N3904) | | Suhu (℃) |
|---|---|---|---|---|
| *Duty Cycle* | Tegangan | *Duty Cycle* | Tegangan | |
| 0% | 4,97 | 0% | 0 | 27,5 |
| 10% | 4,48 | 10% | 1,007 | 29,31 |
| 20% | 3,98 | 20% | 2,14 | 33,5 |
| 30% | 3,491 | 30% | 3,222 | 37,56 |
| 40% | 2,983 | 40% | 4,37 | 39,44 |
| 50% | 2,493 | 50% | 5,47 | 42,75 |
| 60% | 1,989 | 60% | 6,59 | 45,56 |
| 70% | 1,501 | 70% | 7,71 | 49,88 |
| 80% | 0,994 | 80% | 8,82 | 52,94 |
| 90% | 0,508 | 90% | 9,92 | 55,94 |
| 100% | 0,001 | 100% | 11,12 | 59,69 |

Dari tabel 4.3. dapat dilihat bahwa aktuator pemanas dapat menghasilkan suhu pada area permukaan *grill speaker* yang membatasi aktuator pemanas kawat nikrom tidak bersentuhan langsung dengan wadah *stainless steel* bioreaktor mampu menghasilkan suhu maksimum hampir 60℃ ketika diberikan persentase *duty cycle* 100%, dan suhu minimum tidak lain adalah suhu ruangan dalam keadaan normal, tanpa adanya campur tangan pemanasan dari pemanas kawat nikrom (*duty cycle* 0%) didapatkan suhu 27,5℃.




**Gambar 4. 11.** Hubungan Antara Output Nilai PWM Terhadap Suhu yang Dihasilkan




**Gambar 4. 12.** Hubungan Antara Nilai PWM Terhadap Keluaran Tegangan IRF540N

## 4.3 Pengujian *Lookup Table* Konversi PH Menjadi Setpoint Suhu

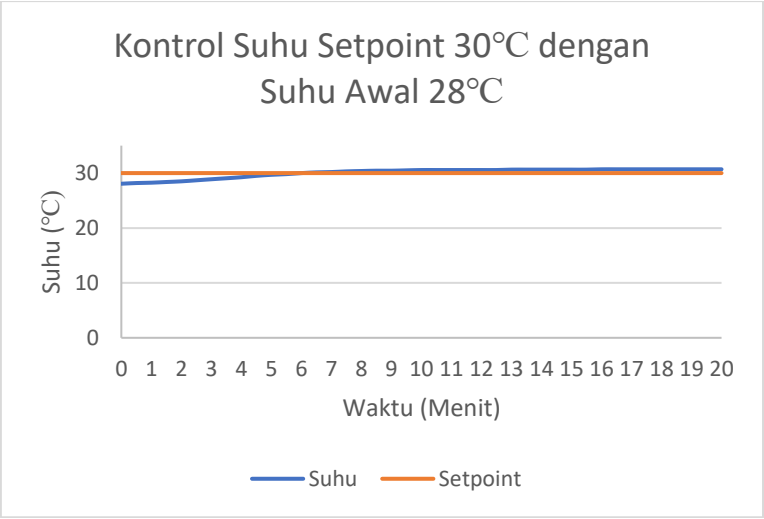
Pada bagian ini dilakukan pengujian hasil konversi tingkat keasaman menjadi setpoint suhu oleh *lookup table* yang telah disusun sebelumnya dengan sampel uji dari input rentang tingkat keasaman 7,5-4,0. Adapun hasil pengujian *lookup table* dari konversi tingkat keasaman menjadi setpoint suhu dapat diamati pada tabel 4.4.

**Tabel 4. 4.** Hasil Konversi *Lookup Table*

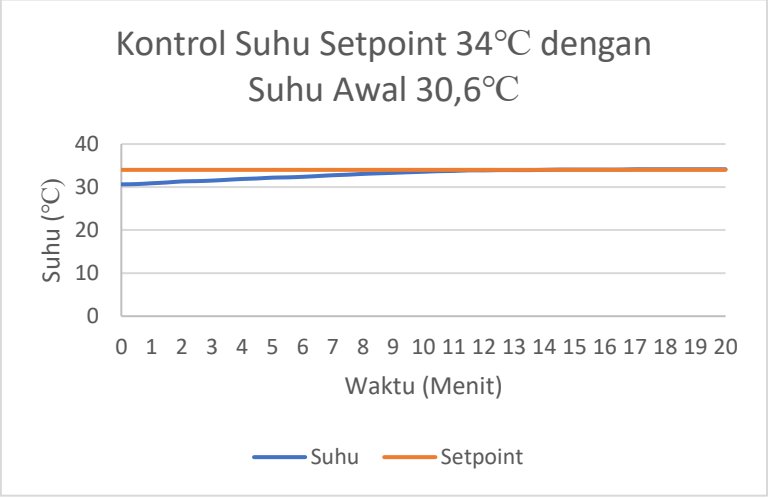
| Tingkat Keasaman (PH) | Setpoint Suhu (℃) |
|---|---|
| 7,5-6,3 | 37 |
| 6,29-5,51 | 30 |
| 5,5-4 | 25 |

## 4.4 Pengujian Sistem Kontrol Logika Fuzzy

Pada subab ini membahas pengujian kesesuaian setpoint suhu dengan media air sebagai pengganti sementara rusip dalam bioreaktor. Pengujian ini sesungguhnya bertugas untuk mengetahui perbandingan suhu setpoint terhadap suhu realita yang terbaca oleh sensor suhu DS18B20. Pengujian suhu menggunakan sampel data sebanyak 3 buah yang bervariasi rentang antar suhu awal dan suhu target kontrol atau setpointnya satu sama lain, dimana sampel pengujian kontrol suhu yang dipilih yaitu kontrol suhu setpoint 30℃ dengan suhu awal 28℃, kontrol suhu setpoint 34℃ dengan suhu awal 30,6℃, dan kontrol suhu setpoint 37℃ dengan suhu awal 30℃. Hasil pengujian yang diperoleh dari masing-masing pengujian akan diolah dalam bentuk tabel dan grafik dengan sampling per sampel data selama 1 menit, secara detail data sampling pengujian yang diperoleh dapat dilihat pada tabel 4.5, 4.6, dan 4.7, sedangkan grafik visualisasi dari korelasi parameter input suhu setpoint terhadap suhu bioreaktor dapat diamati pada gambar 4.13, 4.14, dan 4.15.




**Gambar 4. 13.** Grafik Respon Sistem Kontrol Logika Fuzzy Pada Setpoint Suhu 30℃ dengan Suhu Awal 28℃




**Gambar 4. 14.** Grafik Respon Sistem Kontrol Logika Fuzzy pada Setpoint Suhu 34℃ dengan Suhu Awal 30,5℃

**Tabel 4. 5.** Pengujian Kontrol Logika Fuzzy Pada Setpoint Suhu 30℃ dengan Suhu Awal 28℃

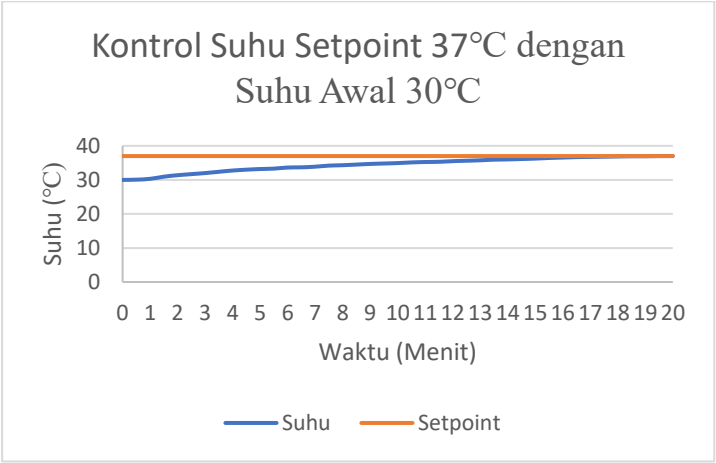
| Menit | Suhu Bioreaktor (℃) |
|---|---|
| 0 | 28,06 |
| 1 | 28,25 |
| 2 | 28,50 |
| 3 | 28,88 |
| 4 | 29,25 |
| 5 | 29,69 |
| 6 | 30 |
| 7 | 30,19 |
| 8 | 30,38 |
| 9 | 30,44 |
| 10 | 30,5 |
| 11 | 30,56 |
| 12 | 30,56 |
| 13 | 30,63 |
| 14 | 30,63 |
| 15 | 30,63 |
| 16 | 30,69 |
| 17 | 30,69 |
| 18 | 30,69 |
| 19 | 30,69 |
| 20 | 30,69 |

**Tabel 4. 6.** Pengujian Kontrol Logika Fuzzy Pada Setpoint Suhu 34℃ dengan Suhu Awal 30,6℃

| Menit | Suhu Bioreaktor (℃) |
|---|---|
| 0 | 30,63 |
| 1 | 30,88 |
| 2 | 31,31 |
| 3 | 31,50 |
| 4 | 31,88 |
| 5 | 32,19 |
| 6 | 32,38 |
| 7 | 32,75 |
| 8 | 33,06 |
| 9 | 33,31 |
| 10 | 33,56 |
| 11 | 33,75 |
| 12 | 33,88 |
| 13 | 33,94 |
| 14 | 34 |
| 15 | 34,06 |
| 16 | 34,06 |
| 17 | 34,13 |
| 18 | 34,13 |
| 19 | 34,13 |
| 20 | 34,13 |

**Tabel 4. 7.** Pengujian Kontrol Logika Fuzzy pada Setpoint Suhu 37℃ dengan Suhu Awal 30℃

| Menit | Suhu Bioreaktor |
|---|---|
| 0 | 30 |
| 1 | 30,31 |
| 2 | 31,38 |
| 3 | 32 |
| 4 | 32,75 |
| 5 | 33,19 |
| 6 | 33,63 |
| 7 | 33,88 |
| 8 | 34,31 |
| 9 | 34,69 |
| 10 | 34,94 |
| 11 | 35,25 |
| 12 | 35,5 |
| 13 | 35,75 |
| 14 | 36 |
| 15 | 36,25 |
| 16 | 36,56 |
| 17 | 36,75 |
| 18 | 36,88 |
| 19 | 36,94 |
| 20 | 37 |




**Gambar 4. 15.** Grafik Respon Sistem Kontrol Logika Fuzzy Pada Setpoint Suhu 37℃ dengan Suhu Awal 30℃

Akan tetapi jika dilihat secara seksama penurunan suhu menuju setpoint tidak secepat seperti menaikkannya hal ini dikarenakan pengaruh desain bioreaktor yang rapat dan dilengkapi dengan casing pelindung kayu dan sterofoam yang tergolong isolator panas yang baik. Oleh karenanya *losses* panas lingkungan sekitar lingkungan bioreaktor tidak cepat tersebar atau diserap oleh udara yang lebih dingin. Ditambah lagi terkadang ruangan percobaan dalam keadaan hangat terlebih saat di siang hari karena terkena paparan langsung dari panas matahari yang merupakan faktor utama yang mempengaruhi suhu ruangan menjadi lebih hangat meskipun pengujian telah dilakukan dalam keadaan ruangan tertutup dengan pendingin udara atau AC menyala.

## 4.5 Pengujian Sistem Kontrol Keseluruhan

Pengujian sistem kontrol keseluruhan dilakukan untuk menganalisa dan mengevaluasi sistem kontrol secara keseluruhan, Menilai rancangan setiap subsistem beroperasi dengan baik, atau sebaliknya tidak beroperasi sesuai tugasnya. pengujian dalam subab ini secara garis besar meliputi pengujian aktuator yang terintegrasi

elemen umpan balik sensor suhu DS18B20 dan sensor pH baik itu dengan metode konversi *lookup table* saja maupun *lookup table* teroptimalisasi kontrol logika fuzzy, hasil interface tampilan parameter kontrol tertentu, dan terakhir pengujian data *logging* dari komponen modul SD Card yang menggambil data *timestamp* dari modul RTC DS3231. Wujud fisik keseluruhan sistem yang dikerjakan dalam tugas akhir ini dapat diamati pada gambar 4.16.

**Gambar 4. 16.** Realisasi Keseluruhan Sistem

Berbeda dari subab lainnya, seluruh rangkaian pengujian sistem kontrol keseluruhan dilaksanakan menggunakan rusip yang berbahan baku utama ikan teri (bilis), garam, dan gula merah seperti ditampilkan pada gambar 4.17, dan 4.18.

Untuk hasil sampling data pengujian rusip dilaksanakan dengan sampling data diambil secara acak selama selama 30 menit, sampling waktu per menit dan data yang diambil adalah data fermentasi hari ke 2, 5, dan 8. Ketiga data pengujian tersebut dapat dilihat pada tabel 4.9, 4.10, dan 4.11, serta perubahan suhu dan tingkat keasaman per hari dapat diamati pada gambar 4.19, dan 4.20.

Dari tabel 4.8 terlihat bahwa terdapat dua buah sampel pengukuran yang terlalu jauh simpangannya. Apabila dibandingkan dengan data pengukuran lainnya, yaitu saat waktu 22.38 dengan pH pengukuran sebesar 6,02 merupakan nilai pengukuran pH terkecil, dan

saat waktu 22.49 dengan pH pengukuran sebesar 6,57 merupakan nilai pengukuran tertinggi, sehingga pembacaan pH kedua besaran tersebut terlihat tidak akurat.

**Tabel 4. 8.** Rata-Rata Data Pengukuran Fermentasi Hari Kedua

| Waktu (Menit) | PH Pengukuran | Suhu Pengukuran | Setpoint Suhu | PWM |
|---|---|---|---|---|
| 22.30 | 6,38 | 37,06 | 37 | 8 |
| 22.31 | 6,24 | 37,06 | 37 | 8 |
| 22.32 | 6,28 | 37,06 | 37 | 8 |
| 22.33 | 6,35 | 37,06 | 37 | 8 |
| 22.34 | 6,32 | 37,06 | 37 | 8 |
| 22.35 | 6,29 | 37,06 | 37 | 8 |
| 22.36 | 6,33 | 37,06 | 37 | 8 |
| 22.37 | 6,33 | 37,06 | 37 | 8 |
| 22.38 | 6,02 | 37,06 | 30 | 8 |
| 22.39 | 6,40 | 37,06 | 37 | 8 |
| 22.40 | 6,29 | 37,06 | 30 | 8 |
| 22.41 | 6,16 | 37,06 | 30 | 8 |
| 22.42 | 6,32 | 37,13 | 37 | 8 |
| 22.43 | 6,38 | 37,13 | 37 | 8 |
| 22.44 | 6,30 | 37,13 | 37 | 8 |
| 22.45 | 6,22 | 37,13 | 30 | 8 |
| 22.46 | 6,30 | 37,13 | 37 | 8 |
| 22.47 | 6,34 | 37,13 | 37 | 8 |
| 22.48 | 6,38 | 37,13 | 37 | 43 |
| 22.49 | 6,57 | 37,19 | 37 | 9 |
| 22.50 | 6,34 | 37,19 | 37 | 9 |
| 22.51 | 6,30 | 37,19 | 37 | 40 |
| 22.52 | 6,22 | 37,19 | 30 | 9 |
| 22.53 | 6,30 | 37,19 | 37 | 9 |
| 22.54 | 6,18 | 37,19 | 30 | 8 |
| 22.55 | 6,21 | 37,19 | 30 | 9 |
| 22.56 | 6,24 | 37,19 | 30 | 9 |
| 22.57 | 6,32 | 37,19 | 37 | 9 |
| 22.58 | 6,29 | 37,19 | 30 | 9 |
| 22.59 | 6,32 | 37,19 | 37 | 9 |

Adanya ketidakakuratan pengukuran pH disebabkan oleh bentuk rusip yang berwujud semi padat yang membuat sensor pH kurang mendapat cairan pengukuran dan pengaruh suhu bioreaktor tidak berada pada keadaan suhu ruangan atau suhu semakin tinggi dapat mempengaruhi pengukuran pH itu sendiri. Sebaliknya keluaran PWM dinilai sudah sesuai dimana menghasilkan nilai *duty cycle* kecil dan pada waktu menit tertentu sedikit mengeset besaran 40 hingga 47 guna mempertahankan suhu pengukuran sesuai dengan suhu setpoint, meskipun berdampak terlihat sedikit naik namun masih dalam rentang *steady state* perancangan sistem kontrol logika fuzzy. Selain itu untuk konversi tingkat keasaman menjadi suhu setpoint terihat sudah sesuai dengan besar konversi yang tertera dalam desain penyusunan isi *lookup table*.

**Gambar 4. 17.** Ikan Teri Bahan Baku Utama Rusip

(a) (b)

**Gambar 4. 18.** Irisan Gula Merah pada Bagian (a) dan Garam Meja pada Bagian (b)

**Tabel 4. 9. Rata-Rata Data Pengukuran Fermentasi Hari Kelima**

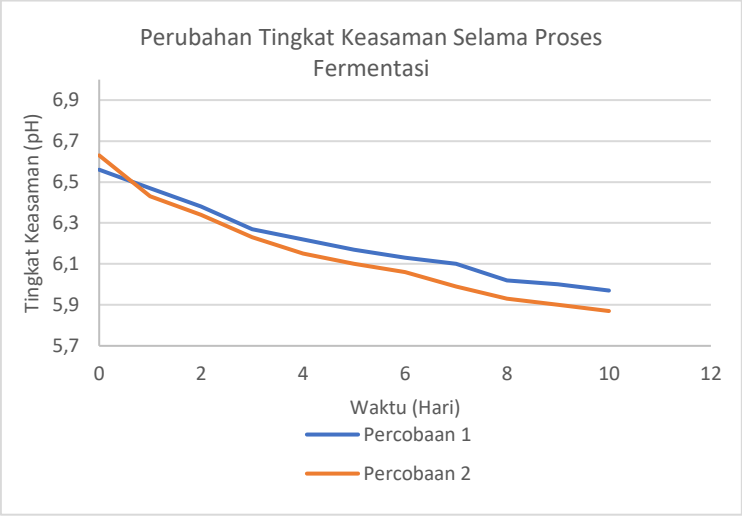
| Waktu (Menit) | PH Pengukuran | Suhu Pengukuran | Setpoint Suhu | PWM |
|---|---|---|---|---|
| 06.15 | 5.85 | 36,94 | 30 | 8 |
| 06.16 | 6.17 | 37 | 30 | 8 |
| 06.17 | 6,13 | 37 | 30 | 8 |
| 06.18 | 6.44 | 37 | 37 | 60 |
| 06.19 | 6.37 | 37 | 37 | 60 |
| 06.20 | 5.97 | 37 | 30 | 8 |
| 06.21 | 6.13 | 37 | 30 | 8 |
| 06.22 | 6.11 | 37 | 30 | 8 |
| 06.23 | 6.09 | 36,94 | 30 | 8 |
| 06.24 | 6,41 | 36,94 | 37 | 60 |
| 06.25 | 6,20 | 36,94 | 30 | 8 |
| 06.26 | 6,40 | 36,94 | 37 | 60 |
| 06.27 | 5,99 | 36,94 | 30 | 8 |
| 06.28 | 6,20 | 36,94 | 30 | 8 |
| 06.29 | 6,18 | 36,94 | 30 | 8 |
| 06.30 | 5,99 | 37 | 30 | 8 |
| 06.31 | 6,74 | 37 | 37 | 51 |
| 06.32 | 6.02 | 37 | 30 | 8 |
| 06.33 | 6.17 | 37 | 30 | 8 |
| 06.34 | 6.29 | 37 | 30 | 8 |
| 06.35 | 6,04 | 37 | 30 | 8 |
| 06.36 | 5,83 | 37 | 30 | 8 |
| 06.37 | 6,21 | 37 | 30 | 8 |
| 06.38 | 6,04 | 37 | 30 | 8 |
| 06.39 | 6,16 | 36,94 | 30 | 8 |
| 06.41 | 6,28 | 36,94 | 30 | 8 |
| 06.42 | 6,22 | 36,94 | 30 | 8 |
| 06.43 | 6,09 | 36,94 | 30 | 8 |
| 06.44 | 6,15 | 36.94 | 30 | 8 |

**Tabel 4. 10.** Rata-Rata Data Pengukuran Fermentasi Hari Kedelapan

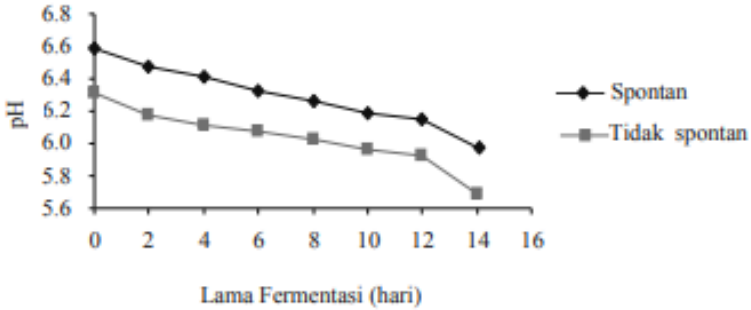
| Waktu (Menit) | PH Pengukuran | Suhu Pengukuran | Setpoint Suhu | PWM |
|---|---|---|---|---|
| 07.00 | 6.04 | 35,38 | 30 | 8 |
| 07.01 | 6.07 | 35,38 | 30 | 8 |
| 07.02 | 6,04 | 35,38 | 30 | 8 |
| 07.03 | 5,96 | 35,38 | 30 | 8 |
| 07.04 | 6,00 | 35,38 | 30 | 8 |
| 07.05 | 6,11 | 35,38 | 30 | 8 |
| 07.06 | 6.01 | 35,38 | 30 | 8 |
| 07.07 | 6.51 | 35,38 | 37 | 230 |
| 07.08 | 6.03 | 35,44 | 30 | 8 |
| 07.09 | 6,34 | 35,44 | 37 | 230 |
| 07.10 | 5,98 | 35,50 | 30 | 8 |
| 07.11 | 6,03 | 35,56 | 30 | 8 |
| 07.12 | 6,05 | 35,63 | 30 | 8 |
| 07.13 | 6,15 | 35,69 | 30 | 8 |
| 07.14 | 6,18 | 35,81 | 30 | 8 |
| 07.15 | 6,21 | 35,94 | 30 | 8 |
| 07.16 | 5,86 | 36 | 30 | 8 |
| 07.17 | 5,97 | 36,06 | 30 | 8 |
| 07.18 | 6,37 | 36,06 | 37 | 230 |
| 07.19 | 5,74 | 36,06 | 30 | 8 |
| 07.20 | 5,85 | 36,13 | 30 | 8 |
| 07.21 | 6,02 | 36,19 | 30 | 8 |
| 07.22 | 5,99 | 36,19 | 30 | 8 |
| 07.23 | 6,04 | 36,19 | 30 | 8 |
| 07.24 | 5,94 | 36,19 | 30 | 9 |
| 07.25 | 6,07 | 36,19 | 30 | 8 |
| 07.26 | 6,34 | 36,19 | 37 | 79 |
| 07.27 | 6,37 | 36,19 | 37 | 79 |
| 07.28 | 6,02 | 36.25 | 30 | 8 |
| 07.29 | 6,04 | 36,31 | 30 | 8 |

Berdasarkan hasil pengujian keseluruhan sistem selama 30 menit secara acak dengan data fermentasi hari 2, 5, dan 8. Terlihat

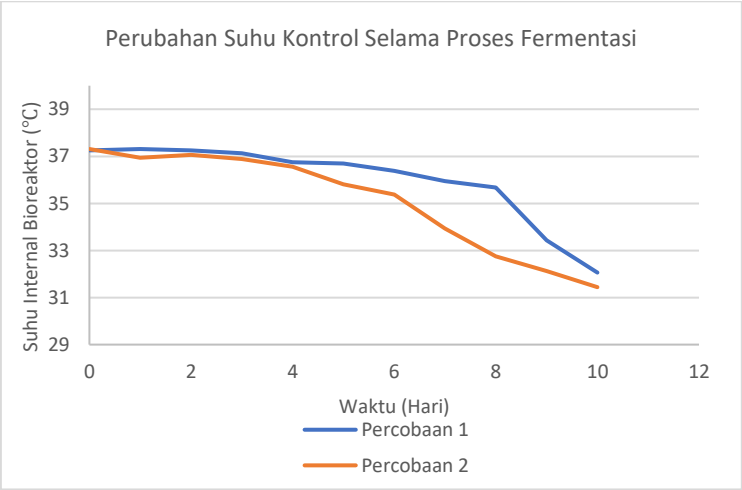
bahwa terdapat error atau kesalahan pada pengukuran tingkat keasaman pada label data yang diberikan warna merah, terjadi kesalahan dengan kesalahan terbesar terjadi ketika suhu bioreaktor sedang bervariasi dan suhu berada pada rentang suhu diatas suhu ruangan antara 35℃ hingga 37℃ dimana memiliki nilai kesalahan pengukuran rata-rata bervariasi 0,2 hingga 0,5 dari data pengukuran sensor tingkat keasaman semestinya, kesalahan pengukuran tingkat keasaman akan berdampak pula pada penerapan hasil nilai pwm pengaturan duty cycle terhadap pemanas, semakin berfluktuasinya suhu bioreaktor akan menyebabkan pengukuran sensor tingkat keasaman semakin tidak akurat dan menurunkan tingkat ketepatan terhadap perulangan pengambilan data pengukuran dengan nilai yang sama atau tingkat presisi sensor tingkat keasmaan.




**Gambar 4. 19.** Grafik Perubahan Harian Tingkat Keasaman Selama Proses Fermentasi Rusip




**Gambar 4. 20.** Grafik Penurunan Tingkat Keasaman dengan Fermentasi Rusip Secara Spontan dan Tidak Spontan [12]




**Gambar 4. 21.** Grafik Perubahan Harian Suhu Selama Proses Fermenasi Rusip

Untuk hasil akhir dari rusip yang telah difermentasi selama 10 hari sesuai dengan target tingkat keasaman yang dikontrol dari pengaturan suhu, bentuk wujud visual produk rusip yang dihasilkan dapat dilihat pada gambar 4.22.

**Gambar 4. 22.** Hasil Akhir Rusip pada Hari ke 10

Adapun hasil pengujian kuisioner *human sensory test* sampel rusip diambil dari 10 orang responden asli asal Bangka Belitung diperoleh melalui pemberian kuisioner setelah mencoba sampel makanan seperti yang ditunjukkan pada gambar 4.23. Untuk keseluruhan penilaian sampel rusip hasil penelitian dibandingkan rusip komersil mendapatkan penilaian rata-rata terbanyak adalah sangat menyukainya baik secara visual, aroma, rasa dan teksturnya seperti ditunjukkan pada grafik 4.24.

**Kuisioner Human Sensory Test Sampel Rusip**

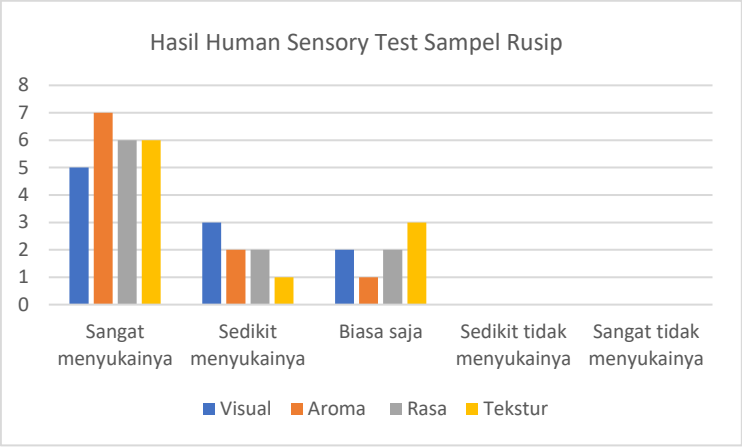
Di depan anda terdapat satu sampel makanan rusip. Berilah penilaian anda mengenai kesesuaian sampel rusip tersebut terhadap rusip yang dijual secara komersil baik dari segi visual, aroma, rasa, dan tekstur dengan ✓pada tabel di bawah ini.

| | Visual | Aroma | Rasa | Tekstur |
|---|---|---|---|---|
| Sangat menyukainya | | | | |
| Sedikit menyukainya | ✓ | ✓ | ✓ | ✓ |
| Biasa saja | | | | |
| Sedikit tidak menyukainya | | | | |
| Sangat tidak menyukainya | | | | |

**Gambar 4. 23.** Kuisioner *Human Sensory Test*

Berikut ini adalah cara perhitungan persentase percepatan dari fermentasi sistem kontrol terhadap fermentasi secara spontan ditinjau dari tingkat keasaman ketika fermentasi spontan telah sesesai dapat diamati pada persamaan (4.1).

$$\%Percepatan\ Proses\ Fermentasi = \frac{Waktu\ Spontan\ - Waktu\ Kontrol}{Waktu\ Spontan} x10 = \frac{14-10}{10} x100\% = 28{,}57\% \quad (4.1)$$




**Gambar 4. 24.** Data Keseluruhan Penilaian *Human Sensory Test*

## 4.6 Analisa Keseluruhan Sistem

Sistem Kontrol Suhu pada Proses Fermentasi Rusip Menggunakan Metode Logika Fuzzy Berdasarkan Tingkat Keasaman ini terdiri dari sensor suhu DS18B20, sensor pH, Arduino Mega, pemanas kawat nikrom, rangkaian *driver* pemanas tersusun atas komponen MOSFET IRF540N dan 2N3904, Arduino Uno, dan LCD TFT. Dari komponen-komponen bagian penyusun yang telah disebutkan baik sensor, aktuator, mikrokontroler, dan perangkat interface terbagi menjadi dua subsistem utama yakni subsistem

kontrol dan subsistem *interface* yang kedua subsistem berkomunikasi secara sederhana dengan protokol komunikasi serial.

Secara keseluruhan, sistem yang telah dirancang dapat bekerja dengan baik, dimana sistem mampu menjaga suhu bioreaktor berada pada kisaran tidak jauh berbeda dengan nilai penentuan setpoint. Hal tersebut sesuai dengan desain awal kontrol logika fuzzy yang dirancang pada tugas akhir ini. Apabila suhu mengalami penurunan atau menjadi dingin dibawah pengaturan setpoint suhu, maka mikrokontroler Arduino mega akan memberikan keluaran gelombang PWM yang berkolerasi sesuai kebutuhan panas dari besar lama kerja atau *duty cycle* pemanas nikrom guna mencapai titik setpoint suhu kembali, dan ketika setpoint suhu telah tercapai maka kawat pemanas nikrom melalui rangkaian *driver* pemanas akan diberikan gelombang PWM bernilai kecil cukup untuk menjaga suhu agar tidak cepat turun dari setpoint suhu. Dan untuk pengaturan suhu berdasarkan tingkat keasaman, sistem terbilang cukup berhasil menyesuaikan besar suhu optimum yang perlu diatur sesuai dengan tingkat keasaman yang terukur oleh sensor pH.

Dalam hal tampilan penyajian data kontrol dan dokumentasi pencatatan data selama proses fermentasi rusip, sistem yang dirancang juga berhasil menampilkan menampilkan seluruh parameter input dan output kontrol dengan baik, serta fitur data *logger* pada sistem mampu merekam data pengukuran sesuai dengan harapan baik itu permenit, jam, maupun harian. Walaupun terkadang terdapat delay antara setiap iterasi siklus mesin dalam mikrokontroler.

Kekurangan dalam sistem yang dirancang setelah dilakukan rangkaian pengujian pertama adalah faktor yang suhu eksternal seperti kondisi ruangan tempat penempatan alat. Apabila suhu ruangan lebih tinggi dari setpoint, maka suhu bioreaktor tidak dapat dengan mudah diturunkan secara spontan. Kekurangan lain ialah tingkat akurasi pengukuran sensor pH yang digunakan dapat menjadi kurang akurat karena dipengaruhi oleh bentuk rusip yang berwujud semi padat, dan dipengaruhi oleh suhu, suhu yang diatas standar suhu ruangan dapat menyebabkan sensor pH menjadi tidak stabil, membuat pengukuran tingkat keasaman menjadi berfluktuasi.

Jika dibangingkan dengan grafik perubahan tingkat keasaman dengan penggunaan metode pengolahan proses awal pembuatan rusip tanpa ditambahkan kultur bakteri asam laktat (spontan) seperti tahapan pembuatan rusip pada jurnal yang diterbitkan oleh D.

Koesoemawardani pada tahun 2013, dimana lampiran grafik penurunan tingkat keasaman tertera pada gambar 4.21. sistem kontrol suhu yang dirancang oleh penulis mampu mempercepat proses fermentasi 28,57% lebih cepat bila dibandingkan fermentasi secara spontan, tanpa sistem kontrol suhu.

Selain itu kelebihan dari penggunaan kontrol logika fuzzy adalah kehandalan sistem menjaga suhu tetap konstan terhadap gangguan yang ada terlebih pada suhu lingkungan yang lebih dingin tanpa mengurangi umur pemakaian pemanas kawat nikrom baik itu akibat *overheating* maupun oksidasi kawat karena perubahan panas yang secara signifikan yang dapat terjadi pada kontrol lain terutama pada on off kontrol.

*Halaman ini sengaja dikosongkan*

# BAB 5
# PENUTUP

## 5.1 Kesimpulan

Berdasarkan pengujian dan analisa sistem kontrol suhu pada proses fermentasi rusip berdasarkan tingkat keasaman, sistem dapat beroperasi dengan baik selama proses fermentasi yang memakan waktu 10 hari tanpa mengalami kendala overheat, tidak hanya itu saja sistem dapat mendeteksi suhu bioreaktor pada rentang suhu fermentasi antara tetap konstan pada rentang 26,38℃-38,31℃, dimana diindikasikan oleh besaran *duty cycle* PWM yang memberikan *duty cycle* PWM keluaran rendah 3,14% - 4%, sedangkan memberikan nilai PWM keluaran menengah 20% - 23,53% ketika suhu pengukuran sesuai dengan setpoint suhu, dan sistem akan memberikan nilai PWM keluaran besar 66,28% - 87,45% ketika suhu pengukuran jauh lebih kecil dari setpoint suhu. Dalam hal penerapan setpoint suhu berdasarkan tingkat keasaman sistem berhasil mengubah seluruh tingkat keasaman sesuai dengan desain penyusunan seperti untuk kisaran tingkat kesaman netral pH >6,3 sistem mampu mengubahnya menjadi setpoint suhu 37℃, untuk kisaran tingkat kesaman sedikit asam pH 6,3-5,9 sistem dapat mengonversikannya menjadi setpoint suhu 30℃, dan kisaran tingkat keasaman asam pH<5,9 sistem dapat mengubahnya menjadi setpoint suhu 25℃.

Perlu diketahui bahwa perangkat yang dikembangkan masih memiliki kelemahan seperti pengukuran tingkat keasaman oleh sensor pH mengalami penurunan akurasi, dan presisi akibat berfluktiasinya data pengukuran apabila digunakan pada keadaan bioreaktor semakin panas di atas rata-rata standar suhu ruangan, serta suhu ruangan dapat mempengaruhi ketepatan suhu kontrol terutama pada saat siang hari atau suhu ruangan yang lebih tinggi dari setpoint suhu yang ingin diatur.

Secara keseluruhan perangkat yang dirancang mampu mempercepatproses fermentasi 28,57% lebih cepat dibandingkan waktu fermentasi rusip komersil dengan bentuk, tekstur, rasa, dan aroma yang tidak jauh berbeda. selain itu perangkat ini dapat juga diterapkan pada produk fermentasi lainnya terutama fermentasi yang berwujud cairan dan menggunakan bakteri asam laktat.

## 5.2 Saran

Saran dari penulis berdasarkan hasil yang diperoleh dari pengujian adalah penambahan sistem pendingin yang dapat membantu proses pengaturan suhu terutama dibawah suhu keadaan ruangan, serta pemilihan sensor pH yang memiliki rentang akurasi lebih sempit dan lebih tahan terhadap faktor suhu seperti suhu lebih panas dan suhu yang bervariasi.

# DAFTAR PUSTAKA


[1] E. Medina, A. de Castro, C. Romero, E. M. Ramírez, dan M. Brenes, "Chapter 18 - Safety of Fermented Fruits and Vegetables," dalam *Regulating Safety of Traditional and Ethnic Foods*, V. Prakash, O. Martín-Belloso, L. Keener, S. Astley, S. Braun, H. McMahon, dan H. Lelieveld, Ed. San Diego: Academic Press, 2016, hlm. 355–367.

[2] O. T. Antonio-Gutiérrez, A. S. López-Díaz, A. López-Malo, E. Palou, dan N. Ramírez-Corona, "7 - UV-C Light for Processing Beverages: Principles, Applications, and Future Trends," dalam *Processing and Sustainability of Beverages*, A. M. Grumezescu dan A. M. Holban, Ed. Woodhead Publishing, 2019, hlm. 205–234.

[3] D. Mergypta, A. Budiharjo, dan E. Kusdiyantini, "Isolasi, Karakterisasi Bakteri Asam Laktat, Dan Analisis Proksimat Dari Pangan Fermentasi Rusip Ikan Teri (Stolephorus sp.)," *J. Akad. Biol.*, vol. 3, no. 2, hlm. 11–19, Mei 2014.

[4] D. Koesoemawardani dan S. Rizal, "Perubahan Sifat Mikrobiologi dan Kimia Rusip dengan Perbedaan Waktu Penambahan Gula Aren Cair," hlm. 8, 2015.

[5] H. Tanguler, "Influence of Temperatures and Fermentation Behaviour of Mixed Cultures of Williopsis saturnus var. saturnus and Saccharomyces cerevisiae Associated with Winemaking," *Food Sci. Technol. Res.*, vol. 19, no. 5, hlm. 781–793, 2013.

[6] Katz, S.E., *The Art of Fermentation: An In-depth Exploration of Essential Concepts and Process from Around the World*, Vermont: Chelsea Green Publishing, 2012.

[7] Azokpota, P.S., "Quality Aspects of Alkaline-Fermented Foods. In: Sarkar, P.K., Nout, M.J.R. (Eds.)," *Handbook of Indigenous Foods Involving Alkaline Fermentation*, CRC Press Taylor & Francis Group, Boca Raton, FL, 2015, hlm. 315-379.

[8] P. F. Stanbury, A. Whitaker, dan S. J. Hall, *Principles of fermentation technology 3rd Edition*, Reprint. Oxford: Butterworth-Heinemann, 2016.

[9] F. Dalcanton, E. Carrasco, F. Pérez-Rodríguez, G. D. Posada-Izquierdo, G. M. Falcão de Aragão, dan R. M. García-Gimeno, "Modeling the Combined Effects of Temperature, pH, and Sodium Chloride and Sodium Lactate Concentrations on the Growth Rate of Lactobacillus plantarum ATCC 8014," *J. Food*

*Qual.*, vol. 2018, hlm. 1726761, Feb 2018, doi: 10.1155/2018/1726761.

[10] "Sambal Rusip, 'Ikan Fermentasi' yang Menggoda - Situs Budaya Indonesia." [Daring]. Tersedia pada: https://www.indonesiakaya.com/jelajah-indonesia/detail/sambal-rusip-ikan-fermentasi-yang-menggoda. [Diakses: 29-Okt-2019].

[11] H. E. Irianto, *Produk Fermentasi Ikan*, Jakarta: Penebar Swadaya Grup, 2013.

[12] D. Koesoemawardani, S. Rizal, dan M. Tauhid, "Perubahan Sifat Mikrobiologi dan Kimiawi Rusip Selama Fermentasi," *J. Agritech*, vol. 3, no. 3, Agustus 2013.

[13] S. Karastogianni, S. Girousi, dan S. Sotiropoulos, "pH: Principles and Measurement," *Encycl. Food Health*, Des 2016.

[14] "The Science Ph Scale | Download now free vectors on Freepik," *Freepik*. [Daring]. Tersedia pada: https://www.freepik.com/free-vector/science-ph-scale_2588946.htm. [Diakses: 29-Okt-2019].

[15] O. Apriyah dan a, "Prinsip Kerja pH Meter," *Artikel Teknologi Indonesia*, 23-Jan-2014.

[16] L. Riadhi, M. Rivai, dan F. Budiman, "Sistem Pengaturan Oksigen Terlarut Menggunakan Metode Logika Fuzzy Berbasis Mikrokontroler Teensy Board," *J. Tek. ITS*, vol. 6, Jul 2017.

[17] M. Rivai, K. Sambodho, dan D. Purnomo, "Identification of Levee Strenght for Early Warning System Using Fuzzy Logic," dalam *International Conference on Information, Communication Technology, and System*, 2014.

[18] "eMathTeacher: Mamdani's fuzzy inference method - Membership functions." [Daring]. Tersedia pada: http://www.dma.fi.upm.es/recursos/aplicaciones/logica_borrosa/web/fuzzy inferencia/funpert_en.htm. [Diakses: 03-Nov-2019].

[19] Abi Nawang Gustica, Muhammad Rivai, Tasripan, "Implementasi Sensor Gas pada Kontrol Lengan Robot untuk Mencari Sumber Gas", Jurnal Teknik POMITS, vol. 3, No. 1, hlm. F.90-95, 2014.

[20] Dewi Nurhaji Meivita, Muhammad Rivai, Astria Nur Irfansyah, "Development of an Electrostatic Air Filtration System Using Fuzzy Logic Control", International Journal on Advanced Science, Engineering and Information Technology, vol. 8, no.4, hlm. 1284-1289, 2018.

[21] K. Sugiarto, M. Rivai, dan A. N. Irfansyah, “Control of Livestock Waste Odors Using Gas Sensors and Fuzzy Logic,” 2019, hlm. 81–86.
[22] K. Jain dan A. Soni, “Comparision of Mamdani and Sugeno Fuzzy Inference System for Deciding the Set Point for a Hydro Power Plant Dam Reservoir Based on Power Generation Requirement,” *Int. J. Eng.*, vol. 2, no. 2, hlm. 4, 2015.
[23] D. J. Hemath, D. Gupta, dan V. E. Balas, *Intelligent Data Analysis for Biomedical Applications*. Oxford: Academic Press, 2019.
[24] R. Bělohlávek, J. W. Dauben, dan G. J. Klir, *Fuzzy Logic and Mathematics*: *A Historical Perspective*, New York: Oxford University Press, 2017.
[25] “Perbandingan Kualitas Antar Sensor Suhu dengan menggunakan Arduino Pro Mini,” *J. Narodroid*, vol. 12, Jul 2016.
[26] Aditya, “Cara Menggunakan Sensor Suhu Digital DS18B20 di Raspberry Pi,” *Narin Laboratory*, 30-Mei-2016. [Daring]. Tersedia pada: https://tutorkeren.com/artikel/cara-menggunakan-sensor-suhu-digital-ds18b20-di-raspberry-pi.htm. [Diakses: 03-Nov-2019].
[27] Sarmidi, Rahmat, S.I., “Sistem Peringatan Dini Banjir Menggunakan Sensor Ultrasonik Berbasis Arduino Uno,” *Jumantaka*, vol. 2, no. 1, 2019.
[28] “TFT LCD display or monitor.” [Daring]. Tersedia pada: http://www.maxipedia.com/TFT+LCD+display+monitor+panel. [Diakses: 26-Nov-2019].
[29] “Arduino Uno R3 | ATmega328P Microcontroller Board,” *Module143*. [Daring]. Tersedia pada: https://www.module143.com/arduino-uno-r3 atmega328p-microcontroller-board. [Diakses: 04-Jun-2020].
[30] A. Aqeel, “Introduction to Arduino Uno,” *The Engineering Projects*, Jun 21, 2018. [Daring]. Tersedia pada: https://www.theengineeringprojects.com/2018/06/introduction-to-arduino-uno.html. [Diakses: 05-Jun-2020].
[31]“Arduino Uno Pin Diagram, Specifications, Pin Configuration & Programming.” [Daring]. Tersedia pada: https://components101.com/microcontrollers/arduino-uno. [Diakses: 05-Jun-2020].

[32]"Arduino Reference." [Daring]. Tersedia pada: https://www.arduino.cc/reference/en/language/functions/communication/serial/. [Diakses: 05-Jun-2020].
[33]"Arduino Uno For Beginners - Projects, Programming and Parts (Tutorial)," *Makerspaces.com*, Feb 07, 2017. [Daring]. Tersedia pada: https://www.makerspaces.com/arduino-uno-tutorial beginners/. [Diakses: 05-Jun-2020].
[34] "Arduino Mega 2560 Rev3." [Daring]. Tersedia pada: https://store.arduino.cc/usa/mega-2560-r3. [Diakses: 23-Mei-2020].
[35]"MegaPinOut.png (1240×1753)." [Daring]. Tersedia pada: http://yourduino.com/docs/MegaPinOut.png. [Diakses: 13-Mei-2020].
[36]"Arduino DS3231 RTC Module Tutorial - Real Time Clock," *Electronics Hub*, Agu 29, 2018. [Daring]. Tersedia pada: https://www.electronicshub.org/arduino-ds3231-rtc-module-tutorial/. [Diakses: 15-Mei-2020].
[37]"What is a data logger? - Advantages of data logging." [Daring]. Tersedia pada: https://www.omega.co.uk/prodinfo/dataloggers.html. [Diakses: 20-Mei-2020].
[38]"In-Depth Tutorial to Interface Micro SD Card Module with Arduino," *Last Minute Engineers*, Jul 02, 2018. [Daring]. Tersedia pada: https://lastminuteengineers.com/arduino-micro-sd-card-module-tutorial/. [Diakses: 20-Mei-2020].
[39]"IRF540N Pinout, Features, Equivalent & Datasheet." [Daring]. Tersedia pada: https://components101.com/irf540n-pinout-equivalent-datasheet. [Diakses: 21-Mei-2020].
[40]"2N3904 Transistor Pinout, Equivalent, Features & Datasheet." [Daring]. Tersedia pada: https://components101.com/2n3904-pinout-datasheet. [Diakses: 22-Mei-2020].
[41] Kanna, A.R, Chandran, A., Nair, A.S., Arun, R., Benedict, J.J, "Comparison of Calorific Values of Various Fuels from Different Fuel Stations", *Research Inventy: International Journal of Engineering And Science*, vol. 4, hlm. 7-10, April 2017.
[42]"Unistrand 125g Reel 26SWG Nichrome Wire." [Daring]. Tersedia pada: https://www.rapidonline.com/unistrand-125g-reel-26swg-nichrome-wire-05-0520. [Diakses: 24-Mei-2020].

## LAMPIRAN A
### *Program Pembacaan Pengukuran Sensor dan Data Logger*

```
#include "Arduino.h"
#include "MovingAverage.h"
#include <OneWire.h>
#include <DallasTemperature.h>
#include <Wire.h>
#include <RTClib.h>
#include <SPI.h>
#include <SD.h>
RTC_DS3231 rtc;

#define ONE_WIRE_BUS 2
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);

const int analogInPin = A0;
int sensorValue = 0;
unsigned long int avgValue;
int buf[10];

uint32_t runTime = -99999;                    // time for next update
float last_error, error, Derror, presentvalue, setpoint, newSp, nilaipH,
temporary1, temporary2;
float tableTest[60];
MovingAverage <float> test(30);

void setup()
{
 Serial.begin(9600);
 sensors.begin();
 Wire.begin();
 rtc.begin();
 //rtc.adjust(DateTime(F(__DATE__),F(__TIME__)));
 //Serial.print("Initializing SD card...");
 if (!SD.begin(53))
 {
```

```
   Serial.println("initialization failed!");
   return;
 }
 Serial.println("initialization done.");
}

void loop()
{
 static char temp[5];
 static char pH[5];
 static char set[5];
 static char pwm[5];
 DateTime now = rtc.now();
 // This allows for the update of variables for time or accessing the
individual elements.
 sensors.requestTemperatures();
 presentvalue = sensors.getTempCByIndex(0);
if(presentvalue==-127.0)
{
  presentvalue = temporary1;
}
dtostrf(presentvalue, 4, 2, temp);
dtostrf(nilaipH, 3, 2, pH);
dtostrf(setpoint, 4, 2, set);
dtostrf(outputPWM, 5, 2, pwm);
for(int i=0;i<10;i++)
{
  buf[i]=analogRead(analogInPin);
  delay(10);
}
for(int i=0;i<9;i++)
{
 for(int j=i+1;j<10;j++)
 {
  if(buf[i]>buf[j])
  {
   temporary1=buf[i];
   buf[i]=buf[j];
   buf[j]=temporary1;
```

**LAMPIRAN B**

***Program Utama Sistem Kontrol Logika Fuzzy dan Aktuator Pemanas Kawat Nikrom***

```
#include <Fuzzy.h>

const int heaterPin = 6;
float last_error, error, Derror, presentvalue, setpoint, newSp, nilaipH;
//presentvalue adalah datasuhu
int outputPWM;

//Inisialiasi sebuah objek Fuzzy
Fuzzy *fuzzy = new Fuzzy();

//Inisialisasi objek FuzzySet
  FuzzySet *errorn = new FuzzySet(-50,-50,-0.5,0);
  //Inisialisasi objek FuzzySet
  FuzzySet *errorz = new FuzzySet(-0.5,0,0,0.5);
  //Inisialisasi objek FuzzySet
  FuzzySet *errorps = new FuzzySet(0,2,2,5);
  //Inisialisasi objek FuzzySet
  FuzzySet *errorpm = new FuzzySet(3,15,15,25);
  //Inisialisasi objek FuzzySet
  FuzzySet *errorpl = new FuzzySet(15,30,50,50);

  //Inisialisasi objek FuzzySet
  FuzzySet *Derrornl = new FuzzySet(-5,-5,-3,-1.5);
  //Inisialisasi objek FuzzySet
  FuzzySet *Derrornm = new FuzzySet(-2.5,-1.5,-1.5,-0.5);
  //Inisialisasi objek FuzzySet
  FuzzySet *Derrorns = new FuzzySet(-1,-0.5,-0.5,0);
  //Inisialisasi objek FuzzySet
  FuzzySet *Derrorz = new FuzzySet(-0.2,0,0,0.2);
  //Inisialisasi objek FuzzySet
  FuzzySet *Derrorps = new FuzzySet(0,0.5,0.5,1);
  //Inisialisasi objek FuzzySet
  FuzzySet *Derrorpm = new FuzzySet(0.5,1.5,1.5,2.5);
  //Inisialisasi objek FuzzySet
  FuzzySet *Derrorpl = new FuzzySet(1.5,3,5,5);
```

```
  //Inisialisasi objek FuzzySet
  FuzzySet *asam = new FuzzySet(5,5,5.5,5.7);
  //Inisialisasi objek FuzzySet
  FuzzySet *sedikitasam = new FuzzySet(5.5,5.7,5.9,6.3);
  //Inisialisasi objek FuzzySet
  FuzzySet *netral = new FuzzySet(5.9,6.3,7.5,7.5);

  //Inisialisasi objek FuzzySet
  FuzzySet *pwm0 = new FuzzySet(0,0,0,25.5);
  //Inisialisasi objek FuzzySet
  FuzzySet *pwm10 = new FuzzySet(0,25.5,25.5,51);
  //Inisialisasi objek FuzzySet
  FuzzySet *pwm20 = new FuzzySet(25.5,51,51,76.5);
  //Inisialisasi objek FuzzySet
  FuzzySet *pwm30 = new FuzzySet(51,76.5,76.5,102);
  //Inisialisasi objek FuzzySet
  FuzzySet *pwm50 = new FuzzySet(76.5,127.5,127.5,178.5);
  //Inisialisasi objek FuzzySet
  FuzzySet *pwm70 = new FuzzySet(127.5,178.5,178.5,229.5);
  //Inisialisasi objek FuzzySet
  FuzzySet *pwm100 = new FuzzySet(178.5,255,255,255);

  //Inisialisasi objek FuzzyOutput
  FuzzySet *leuconostoc = new FuzzySet(20,20,20,35);
  //Inisialisasi objek FuzzyOutput
  FuzzySet *lactococcus = new FuzzySet(25,30,30,35);
  //Inisialisasi objek FuzzyOutput
  FuzzySet *streptococcus = new FuzzySet(30,37,37,45);

void MembershipFunc()
{
  //Inisialisasi objek FuzzyInput
  FuzzyInput *inputerror = new FuzzyInput(1);
  //Menggabungkan FuzzySet penyusun membership fuction
FuzzyInput
  inputerror->addFuzzySet(errorn);
  inputerror->addFuzzySet(errorz);
  inputerror->addFuzzySet(errorps);
```

```
  inputerror->addFuzzySet(errorpm);
  inputerror->addFuzzySet(errorpl);
  //Mengaktifkan membership fuction FuzzyInput
  fuzzy->addFuzzyInput(inputerror);

  //Inisialisasi objek FuzzyInput
  FuzzyInput *inputDerror = new FuzzyInput(2);
  //Menggabungkan FuzzySet penyusun membership function
FuzzyInput
  inputDerror->addFuzzySet(Derrornl);
  inputDerror->addFuzzySet(Derrornm);
  inputDerror->addFuzzySet(Derrorns);
  inputDerror->addFuzzySet(Derrorz);
  inputDerror->addFuzzySet(Derrorps);
  inputDerror->addFuzzySet(Derrorpm);
  inputDerror->addFuzzySet(Derrorpl);
  //Mengaktifkan membership function FuzzyInput
  fuzzy->addFuzzyInput(inputDerror);

  //Inisialisasi objek FuzzyInput
  FuzzyInput *inputpH = new FuzzyInput(3);
  //Menggabungkan FuzzySet penyusun membership function
FuzzyInput
  inputpH->addFuzzySet(asam);
  inputpH->addFuzzySet(sedikitasam);
  inputpH->addFuzzySet(netral);
  //Mengkatifkan membership function FuzzyInput
  fuzzy->addFuzzyInput(inputpH);

  //Inisialisasi objek FuzzyOutput
  FuzzyOutput *outputpwmHeater = new FuzzyOutput(1);
  //Menggabungkan FuzzySet penyusun membership function
FuzzyOutput
  outputpwmHeater->addFuzzySet(pwm0);
  outputpwmHeater->addFuzzySet(pwm10);
  outputpwmHeater->addFuzzySet(pwm20);
  outputpwmHeater->addFuzzySet(pwm30);
  outputpwmHeater->addFuzzySet(pwm50);
  outputpwmHeater->addFuzzySet(pwm70);
```

```
  outputpwmHeater->addFuzzySet(pwm100);
  //Mengaktifkan membership function FuzzyOutput
  fuzzy->addFuzzyOutput(outputpwmHeater);

  //Inisialisasi objek FuzzyOutput
  FuzzyOutput *outputSp = new FuzzyOutput(2);
  //Menggabungkan FuzzySet penyusun membership function FuzzyOutput
  outputSp->addFuzzySet(leuconostoc);
  outputSp->addFuzzySet(lactococcus);
  outputSp->addFuzzySet(streptococcus);
  //Mengaktifkan membership function FuzzyOutput
  fuzzy->addFuzzyOutput(outputSp);
}

void Rule()
{
 //Jika pH = netral maka setpoint suhu = bakteri streptococcus
 //Inisialisasi FuzzyRule bagian jika (if)
 FuzzyRuleAntecedent *ifpHnetral = new FuzzyRuleAntecedent();
 //Menghubungkan FuzzyRule if dengan FuzzySet input tunggal
 ifpHnetral->joinSingle(netral);
 //Inisialisasi FuzzyRule bagian maka (then)
 FuzzyRuleConsequent *thenSpStreptococcus = new FuzzyRuleConsequent();
 //Menghubungkan FuzzyRule then dengan FuzzySet output tunggal
 thenSpStreptococcus->addOutput(streptococcus);
 //Inisialisasi keseluruhan bagian objek FuzzyRule
 FuzzyRule *fuzzyRule01 = new FuzzyRule(1, ifpHnetral, thenSpStreptococcus);
 //Memasukkan FuzzyRule ke dalam Fuzzy
 fuzzy->addFuzzyRule(fuzzyRule01);

 //Jika pH = sedikitasam maka setpoint suhu = bakteri lactococcus
 //Inisialisasi FuzzyRule bagian jika (if)
 FuzzyRuleAntecedent *ifpHsedikitasam = new FuzzyRuleAntecedent();
 //Menghubungkan FuzzyRule if dengan FuzzySet input tunggal
 ifpHsedikitasam->joinSingle(sedikitasam);
```

```
//Inisialisasi FuzzyRule bagian maka (then)
FuzzyRuleConsequent *thenSpLactococcus = new FuzzyRuleConsequent();
//Menghubungkan FuzzyRule then dengan FuzzySet output tunggal
thenSpStreptococcus->addOutput(lactococcus);
//Inisialisasi keseluruhan bagian objek FuzzyRule
FuzzyRule *fuzzyRule02 = new FuzzyRule(2, ifpHsedikitasam, thenSpLactococcus);
//Memasukkan FuzzyRule ke dalam Fuzzy
fuzzy->addFuzzyRule(fuzzyRule02);

//Jika pH = asam maka setpoint suhu = bakteri leuconostoc
//Inisialisasi FuzzyRule bagian jika (if)
FuzzyRuleAntecedent *ifpHasam = new FuzzyRuleAntecedent();
//Menghubungkan FuzzyRule if dengan FuzzySet input tunggal
ifpHasam->joinSingle(asam);
//Inisialisasi FuzzyRule bagian maka (then)
FuzzyRuleConsequent *thenSpLeuconostoc = new FuzzyRuleConsequent();
//Menghubungkan FuzzyRule then dengan FuzzySet output tunggal
thenSpLeuconostoc->addOutput(leuconostoc);
//Inisialisasi keseluruhan bagian objek FuzzyRule
FuzzyRule *fuzzyRule03 = new FuzzyRule(3, ifpHasam, thenSpLeuconostoc);
//Memasukkan FuzzyRule ke dalam Fuzzy
fuzzy->addFuzzyRule(fuzzyRule03);

//Jika error suhu = negatif dan Derror suhu = positif large maka pwm heater = 0%
//Inisialisasi FuzzyRule bagian jika (if)
FuzzyRuleAntecedent *iferrorNegatifDanDerrorPositifLarge = new FuzzyRuleAntecedent();
//Menghubungkan FuzzyRule if dengan dua FuzzySet input dengan operator dan
iferrorNegatifDanDerrorPositifLarge->joinWithAND(errorn, Derrorpl);
//Inisialisasi FuzzyRule bagian maka (then)
FuzzyRuleConsequent *thenpwmHeaterZero1 = new FuzzyRuleConsequent();
```

```
//Menghubungkan FuzzyRule then dengan FuzzySet output tunggal
thenpwmHeaterZero1->addOutput(pwm0);
//Inisialisasi keseluruhan bagian objek FuzzyRule
FuzzyRule *fuzzyRule04 = new FuzzyRule(4, iferrorNegatifDanDerrorPositifLarge, thenpwmHeaterZero1);
//Memasukkan FuzzyRule ke dalam Fuzzy
fuzzy->addFuzzyRule(fuzzyRule04);

//Jika error suhu = negatif dan Derror suhu = positif medium maka pwm heater = 0%
//Inisialisasi FuzzyRule bagian jika (if)
FuzzyRuleAntecedent *iferrorNegatifDanDerrorPositifMedium = new FuzzyRuleAntecedent();
//Menghubungkan FuzzyRule if dengan dua FuzzySet input dengan operator dan
iferrorNegatifDanDerrorPositifMedium->joinWithAND(errorn, Derrorpm);
//Inisialisasi FuzzyRule bagian maka (then)
FuzzyRuleConsequent *thenpwmHeaterZero2 = new FuzzyRuleConsequent();
//Menghubungkan FuzzyRule then dengan FuzzySet output tunggal
thenpwmHeaterZero2->addOutput(pwm0);
//Inisialisasi keseluruhan bagian objek FuzzyRule
FuzzyRule *fuzzyRule05 = new FuzzyRule(5, iferrorNegatifDanDerrorPositifMedium, thenpwmHeaterZero2);
//Memasukkan FuzzyRule ke dalam Fuzzy
fuzzy->addFuzzyRule(fuzzyRule05);

//Jika error suhu = negatif dan Derror suhu = positif small maka pwm heater = 0%
//Inisialisasi FuzzyRule bagian jika (if)
FuzzyRuleAntecedent *iferrorNegatifDanDerrorPositifSmall = new FuzzyRuleAntecedent();
//Menghubungkan FuzzyRule if dengan dua FuzzySet input dengan operator dan
iferrorNegatifDanDerrorPositifSmall->joinWithAND(errorn, Derrorps);
//Inisialisasi FuzzyRule bagian maka (then)
```

```
 FuzzyRuleConsequent *thenpwmHeaterZero3 = new
FuzzyRuleConsequent();
 //Menghubungkan FuzzyRule then dengan FuzzySet output tunggal
 thenpwmHeaterZero3->addOutput(pwm0);
 //Inisialisasi keseluruhan bagian objek FuzzyRule
 FuzzyRule *fuzzyRule06 = new FuzzyRule(6,
iferrorNegatifDanDerrorPositifSmall, thenpwmHeaterZero3);
iferrorPositifLargeDanDerrorNegatifSmall->joinWithAND(errorpl,
Derrorns);
 //Inisialisasi FuzzyRule bagian maka (then)
 FuzzyRuleConsequent *thenpwmHeaterHundred10 = new
FuzzyRuleConsequent();
 //Menghubungkan FuzzyRule then dengan FuzzySet output tunggal
 thenpwmHeaterHundred10->addOutput(pwm100);
 //Inisialisasi keseluruhan bagian objek FuzzyRule
 FuzzyRule *fuzzyRule36 = new FuzzyRule(36,
iferrorPositifLargeDanDerrorNegatifSmall,
thenpwmHeaterHundred10);
 //Memasukkan FuzzyRule ke dalam Fuzzy
 fuzzy->addFuzzyRule(fuzzyRule36);

 //Jika error suhu = positif large dan Derror suhu = negatif medium
maka pwm heater = 100%
 //Inisialisasi FuzzyRule bagian jika (if)
 FuzzyRuleAntecedent
*iferrorPositifLargeDanDerrorNegatifMedium = new
FuzzyRuleAntecedent();
 //Menghubungkan FuzzyRule if dengan dua FuzzySet input dengan
operator dan

iferrorPositifLargeDanDerrorNegatifMedium->joinWithAND(errorp
l, Derrornm);
 //Inisialisasi FuzzyRule bagian maka (then)
 FuzzyRuleConsequent *thenpwmHeaterHundred11 = new
FuzzyRuleConsequent();
 //Menghubungkan FuzzyRule then dengan FuzzySet output tunggal
 thenpwmHeaterHundred11->addOutput(pwm100);
 //Inisialisasi keseluruhan bagian objek FuzzyRule
```

```
 FuzzyRule *fuzzyRule37 = new FuzzyRule(37,
iferrorPositifLargeDanDerrorNegatifMedium,
thenpwmHeaterHundred11);
 //Memasukkan FuzzyRule ke dalam Fuzzy
 fuzzy->addFuzzyRule(fuzzyRule37);

 //Jika error suhu = positif large dan Derror suhu = negatif large maka
pwm heater = 100%
 //Inisialisasi FuzzyRule bagian jika (if)
 FuzzyRuleAntecedent *iferrorPositifLargeDanDerrorNegatifLarge =
new FuzzyRuleAntecedent();
 //Menghubungkan FuzzyRule if dengan dua FuzzySet input dengan
operator dan
 iferrorPositifLargeDanDerrorNegatifLarge->joinWithAND(errorpl,
Derrornl);
 //Inisialisasi FuzzyRule bagian maka (then)
 FuzzyRuleConsequent *thenpwmHeaterHundred12 = new
FuzzyRuleConsequent();
 //Menghubungkan FuzzyRule then dengan FuzzySet output tunggal
 thenpwmHeaterHundred12->addOutput(pwm100);
 //Inisialisasi keseluruhan bagian objek FuzzyRule
 FuzzyRule *fuzzyRule38 = new FuzzyRule(38,
iferrorPositifLargeDanDerrorNegatifLarge,
thenpwmHeaterHundred12);
 //Memasukkan FuzzyRule ke dalam Fuzzy
 fuzzy->addFuzzyRule(fuzzyRule38);
}

void Fuzzifikasi()
{
  //Update nilai setpoint, error dan delta error suhu
  if(nilaipH>=6.30)
  {
    setpoint = 37;
  }
  else if(nilaipH>=5.90 && nilaipH<6.30)
  {
    setpoint = 30;
  }
```

```
  else if(nilaipH<5.90)
  {
    setpoint = 25;
  }
  //Serial.print("Setpoint sekarang: ");
  //Serial.println(setpoint);
  last_error = error;
  error = setpoint - presentvalue;
  Derror = error - last_error;
  //Menampilkan nilai error dan Derror
  //Serial.print("Last Error: ");
  //Serial.println(last_error);
  //Serial.print("Error: ");
  //Serial.println(error);
  //Serial.print("Derror: ");
  //Serial.println(Derror);
  //Input nilai crisp error dan Derror ke dalam Logika Fuzzy
  fuzzy->setInput(1,error);
  fuzzy->setInput(2,Derror);
  fuzzy->setInput(3,nilaipH);
  //Melakukan proses fuzzifikasi
  fuzzy->fuzzify();
}

void setup()
{
 Serial.begin(9600);
 MembershipFunc();
 Rule();
}

void loop()
{
 Fuzzifikasi();
  //print membership function pH
 Serial.print("pH: asam-> ");
 Serial.print(asam->getPertinence());
 Serial.print(", sedikitasam-> ");
 Serial.print(sedikitasam->getPertinence());
```

```
 Serial.print(", netral-> ");
 Serial.print(netral->getPertinence());
 //print membership function error suhu
 Serial.print("Error: errorn-> ");
 Serial.print(errorn->getPertinence());
 Serial.print(", errorz-> ");
 Serial.print(errorz->getPertinence());
 Serial.print(", errorps-> ");
 Serial.println(errorps->getPertinence());
 Serial.print(", errorpm-> ");
 Serial.println(errorpm->getPertinence());
 Serial.print(", errorpl-> ");
 Serial.println(errorpl->getPertinence());
 //print membership function derror suhu
 Serial.print("Derrorn: Derrornl-> ");
 Serial.print(Derrornl->getPertinence());
 Serial.print(", Derrornm-> ");
 Serial.print(Derrornm->getPertinence());
 Serial.print(", Derrorns-> ");
 Serial.println(Derrorns->getPertinence());
 Serial.print(", Derrorz-> ");
 Serial.println(Derrorz->getPertinence());
 Serial.print(", Derrorps-> ");
 Serial.println(Derrorps->getPertinence());
 Serial.print(", Derrorpm-> ");
 Serial.println(Derrorpm->getPertinence());
 Serial.print(", Derrorpl-> ");
 Serial.println(Derrorpl->getPertinence());
  // Running the Defuzzification
 outputPWM = fuzzy->defuzzify(1);
 newSp = fuzzy->defuzzify(2);
 // Printing Output Control
 Serial.println("Hasil: ");
 Serial.print("PWM: ");
 Serial.println(abs(outputPWM-255));
 analogWrite(heaterPin, abs(outputPWM-255));
 Serial.print("Setpoint Suhu Terbaru: ");
 Serial.println(newSp);
delay(1000);
```

}

## LAMPIRAN C

***Program Arduino Uno Sebagai Sistem Interface Tampilan Parameter Input dan Output Sistem Kontrol***

```
#include <Adafruit_GFX.h>    // Adafruit's core graphics library
#include <Adafruit_TFTLCD.h> // Adafruit's hardware-specific library
#include <TouchScreen.h>     //Touchscreen library

//Sense touch trough these pins
#define YP A3  // must be an analog pin
#define XM A2  // must be an analog pin
#define YM 9   // can be a digital pin
#define XP 8   // can be a digital pin

#define TS_MINX 100
#define TS_MINY 120
#define TS_MAXX 920
#define TS_MAXY 940
//Create the touchscreen object
TouchScreen ts = TouchScreen(XP, YP, XM, YM, 300); //(data,data,data,data,sensitivity);

//Some of the tft pins
#define LCD_CS A3
#define LCD_CD A2
#define LCD_WR A1
#define LCD_RD A0
// Optional, used to reset the display
#define LCD_RESET A4

//Create the tft object
Adafruit_TFTLCD tft(LCD_CS, LCD_CD, LCD_WR, LCD_RD, LCD_RESET);

//File myFile;
```

```
// Define some TFT readable colour codes to human readable names
#define BLACK   0x0000
#define BLUE    0x001F
int boxSize = (sizeX / numberOfMarks);
int mark[] = {(boxSize + deviation), ((boxSize * 2) + deviation), ((boxSize * 3) + deviation), ((boxSize * 4) + deviation), ((boxSize * 5) + deviation), ((boxSize * 6) + deviation), ((boxSize * 7) + deviation), ((boxSize * 8) + deviation), ((boxSize * 9) + deviation), ((boxSize * 10) + deviation)};

const int minorSizeY = (originY + 10);
const int minorSizeX = (originX - 10);

float numberSize = (sizeY / 14);
float number[] = {numberSize, (numberSize * 2), (numberSize * 3), (numberSize * 4), (numberSize * 5), (numberSize * 6), (numberSize * 7), (numberSize * 8), (numberSize * 9), (numberSize * 10), (numberSize * 11), (numberSize * 12), (numberSize * 13), (numberSize * 14)};

int numberValueTemp = (graphRangeTemp / 14);
int valTemp[] = {numberValueTemp, (numberValueTemp * 2), (numberValueTemp * 3), (numberValueTemp * 4), (numberValueTemp * 5), (numberValueTemp * 6), (numberValueTemp * 7), (numberValueTemp * 8), (numberValueTemp * 9), (numberValueTemp * 10), (numberValueTemp * 11), (numberValueTemp * 12), (numberValueTemp * 13), graphRangeTemp};

int numberValuepH = (graphRangepH / 14);
int valpH[] = {numberValuepH, (numberValuepH * 2), (numberValuepH * 3), (numberValuepH * 4), (numberValuepH * 5), (numberValuepH * 6), (numberValuepH * 7), (numberValuepH * 8), (numberValuepH * 9), (numberValuepH * 10), (numberValuepH * 11), (numberValuepH * 12), (numberValuepH * 13), graphRangepH};

int currentpage;
uint32_t runTime = -99999;// time for next update
```

```
double reading = 0; // Value to be displayed

void drawGraph()
{
  // draw outline
  tft.fillScreen(BLACK);
  tft.setTextSize(2);
  tft.fillRoundRect(0, 0, 40, 30, 8, BLUE);
  tft.drawRoundRect(0, 0, 40, 30, 8, WHITE);
  tft.setCursor(5, 8);
  tft.print("<-");
  tft.drawLine(originX, originY, (originX + sizeX), originY,
GREEN);
  tft.drawLine(originX, originY, originX, (originY - sizeY), GREEN);
  tft.setTextSize(1);
  tft.setCursor(230,5);
  tft.print("STATUS: ");
  // draw lables
  for(int i = 0; i < numberOfMarks; i++)
  {
    tft.drawLine(mark[i], originY, mark[i], minorSizeY, GREEN);
  }

  // draw numbers
  for(int i = 0; i < 14; i++)
  {
    tft.drawLine(originX, (originY - number[i]), minorSizeX, (originY
- number[i]), GREEN);
  }

  // draw number values
  for(int i = 0; i < 14; i++)
  {
    tft.setCursor((minorSizeX - 30), (originY - number[i] - 5));
    tft.setTextColor(CYAN);
    tft.setTextSize(1);
    tft.println(valTemp[i]);
    tft.setTextColor(YELLOW);
    tft.setCursor((minorSizeX - 15), (originY - number[i] - 5));
```

```
    tft.println(valpH[i]);
  }
}


void Touch()
{
  TSPoint p = ts.getPoint();// Read touchscreen

  pinMode(XM, OUTPUT);
  pinMode(YP, OUTPUT);

  if (currentpage == 0)
  {
    if (p.z > 10 && p.z < 1000)
    {
      if (p.x > 403 && p.x < 525 && p.y > 271 && p.y < 725)
      {
       //Serial.println("pH");
       currentpage = 2;
       tft.fillRoundRect(45, 130, 230, 40, 8, WHITE);   //rgb led
       delay(70);

       tft.fillRoundRect(45, 130, 230, 40, 8, DARKBLUE);//rgb led
       tft.drawRoundRect(45, 130, 230, 40, 8, WHITE);   //rgb led
       tft.setCursor(145, 142);

       tft.print("pH");
       delay(70);

       parampH();
      }

     if (p.x > 563 && p.x < 683 && p.y > 275 && p.y < 750)
     {
       currentpage = 1;

       tft.fillRoundRect(45, 80, 230, 40, 8, WHITE);
       delay(70);
```

```
      tft.fillRoundRect(45, 80, 230, 40, 8, DARKBLUE);
      tft.drawRoundRect(45, 80, 230, 40, 8, WHITE);

      tft.setCursor(92, 92);
      tft.print("Temperature");
      delay(70);

      paramTemp();
    }

    if (p.x > 143 && p.x < 367 && p.y > 215 && p.y < 725)
    {
      //Serial.println("Time Graphic");

      currentpage = 3;

      tft.fillRoundRect(45, 180, 230, 40, 8, WHITE);
      delay(70);

      tft.fillRoundRect(45, 180, 230, 40, 8, DARKBLUE);
      tft.drawRoundRect(45, 180, 230, 40, 8, WHITE);

      tft.setCursor(89, 192);
      tft.print("Time Graphic");
      delay(70);

      drawGraph();
    }
  }
}

if (currentpage == 1)  //Temperature
  {
  if (p.y > 150 && p.x < 875 && p.x > 743 && p.y < 247)
     {
      tft.fillRoundRect(5, 5, 50, 30, 8, WHITE);
      delay(70);
      tft.fillRoundRect(5, 5, 50, 30, 8, BLUE);
```

```
      tft.drawRoundRect(5, 5, 50, 30, 8, WHITE);
      tft.setCursor(15, 12);
      tft.print("<-");
      delay(70);
      tft.fillRoundRect(5, 5, 50, 30, 8, BLACK);
      currentpage = 0;
      drawHome();
     }
  }

if (currentpage == 2)  //pH
  {
    if (p.y > 150 && p.x < 875 && p.x > 743 && p.y < 247)
     {
      tft.fillRoundRect(5, 5, 50, 30, 8, WHITE);
      delay(70);
      tft.fillRoundRect(5, 5, 50, 30, 8, BLUE);
      tft.drawRoundRect(5, 5, 50, 30, 8, WHITE);
      tft.setCursor(15, 12);
      tft.print("<-");
      delay(70);
      tft.fillRoundRect(5, 5, 50, 30, 8, BLACK);
      currentpage = 0;
      drawHome();
     }
  }

if (currentpage == 3)//Time Graphic
{
  if (p.y > 150 && p.x < 875 && p.x > 743 && p.y < 247)
    {
      tft.fillRoundRect(0, 0, 40, 30, 8, WHITE);
      delay(70);
      tft.setTextSize(2);
      tft.fillRoundRect(0, 0, 40, 30, 8, BLUE);
      tft.drawRoundRect(0, 0, 40, 30, 8, WHITE);
      tft.setCursor(5, 8);
      tft.print("<-");
      delay(70);
```

```
      tft.fillRoundRect(0, 0, 40, 30, 8, BLACK);
      currentpage = 0;
      drawHome();
     }
  }
}

void setup()
{
 Serial.begin(9600);
 tft.reset();
void recvWithStartEndMarkers() {
    static boolean recvInProgress = false;
    static byte ndx = 0;
    char startMarker = '<';
    char endMarker = '>';
    char rc;
    while (Serial.available() > 0 && newData == false) {
        rc = Serial.read();
        if (recvInProgress == true) {
            if (rc != endMarker) {
                receivedChars[ndx] = rc;
                ndx++;
                if (ndx >= numChars) {
                    ndx = numChars - 1;
                }
            }
            else {
                receivedChars[ndx] = '\0'; // terminate the string
                recvInProgress = false;
                ndx = 0;
                newData = true;
            }
        }
        else if (rc == startMarker) {
            recvInProgress = true;
        }
    }
}
```

```
void parseData()
{
   // split the data into its parts
   char * strtokIndx; // this is used by strtok() as an index
   strtokIndx = strtok(tempChars," "); // get the first part - the string
   datemp = atof(strtokIndx);      // convert this part to a float
   strtokIndx = strtok(NULL, " ");
   nilaipH = atof(strtokIndx);
   strtokIndx = strtok(NULL, " ");
   setpoint = atof(strtokIndx);
   strtokIndx = strtok(NULL, " ");
   outputPWM = atof(strtokIndx);
}

void showParsedData()
{
   Serial.print("Temperature: ");
   Serial.println(datemp);
   Serial.print("pH: ");
   Serial.println(nilaipH);
   Serial.print("Setpoint: ");
   Serial.println(setpoint);
   Serial.print("PWM: ");
   Serial.println(outputPWM);
}
```

**LAMPIRAN D**
**Datasheet PH Sensor BNC E201**

# PH Sensor E-201-C

Technical Manual Rev 1r0

***The pH sensor Module*** consist of PH sensor also called as PH probe and a Signal Conditioning Board which gives an output which is proportional to the PH Value and can be interfaced directly to any Micro-controller.
The pH sensor components are usually combined into one device called a combination pH electrode. The measuring electrode is usually glass and quite fragile. Recent developments have replaced the glass with more durable solid-state sensors. The preamplifier is a signal-conditioning device. It takes the high-impedance pH electrode signal and changes it into a low impedance signal which the analyzer or transmitter can accept. The preamplifier also strengthens and stabilizes the signal, making it less susceptible to electrical noise.

pH and ORP probes are both used for measuring the acidic intensity of liquid solutions. A pH probe measures acidity on a scale from 0 to 14, with 0 being the most acidic and 14 being the most basic. Similarly, an Oxidation-Reduction Potential (ORP) probe returns a voltage proportional to the tendency of the solution to gain or lose electrons from other substances (which is linked directly to the pH a substance).

**Applications:**

- Can be used as different tester, pH meter or controller
- Suitable for most aquariums, hydroponics, laboratory etc

**General Specifications:**

***Input supply voltage:*** 5V
***Working current:*** 5 – 10mA
***Detection concentration range:*** PH 0 – 14
***Detection range of temperature:*** 0 – 80 degC
***Response Time::*** ≤ 5S
***Stability Time:*** ≤ 60S
***Output:*** Analog
***Power Consumption:*** ≤ 0.5W
***Working Temperature:*** -10 to +50 deg C
***Working Humidity:*** 95%RH (nominal humidity 65%RH)
***Weight:*** 25g
***PCB Dimension:*** 42mm x 32mm x 20mm


©2017 by e-Gizmo Mechatronix Central

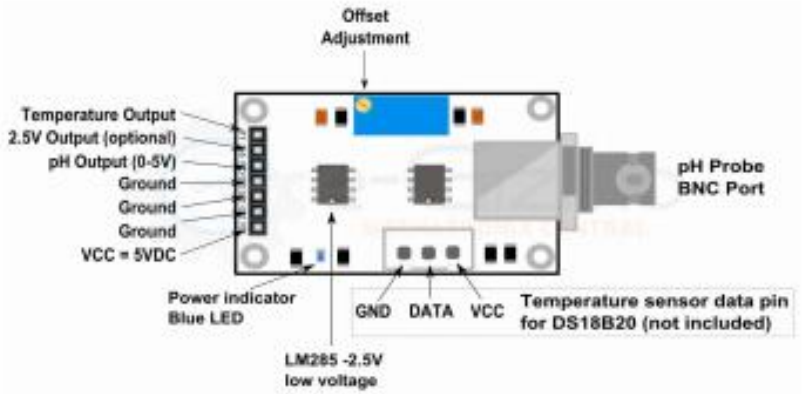



**Figure 1: pH Sensor module kit v2 pinouts**

**Figure 2: pH Subtance (solutions for testing reference)**


©2017 by e-Gizmo Mechatronix Central

***IMPORTANT TIPS:***

**On How to Clean pH Electrodes:**

1. Do not "wipe" or rub the electrode.
2. Swirl the electrode gently in the cleaning soultion.
3. Gently rinse with deionized or distilled water.
4. Store in a storage solution.
5. When possible, use a specialized electrode.

**Which Cleaning Solution to use?**

The cleaning solution you use will depend upon your particular process and the residues you are trying to remove. There is a wide range of pre-mixed cleaning solutions available online or you can make your own. Make sure you take care when handling any cleaning solution – some can be hazardous so make sure you follow all safety instructions and wear appropriate protection equipment!

(For more information - Please go to reference link)

**Reference:**
**https://www.southforkinst.com/ph-electrodes-clean-repair/**

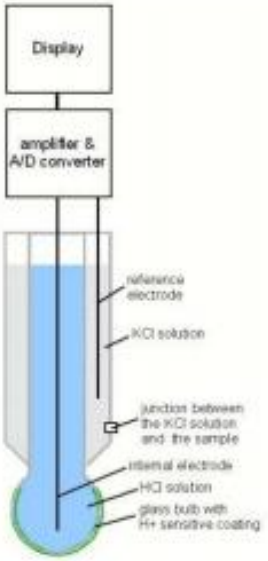



**Figure 3: pH Electrode parts**


©2017 by e-Gizmo Mechatronix Central

**Figure 1: Reference pH Value and Output voltage.**

| pH Value | Output |
|---|---|
| 4 | 3.071 |
| 7 | 2.535 |
| 10 | 2.066 |

*Table 2. pH Value of the known substances.*

| | |
|---|---|
| 0 | 5% Sulphuric acid, H2SO4, battery acid. |
| 1 | 0.1 N HCl, hydrochloric acid (1.1) |
| 2 | Lemon juice. Vinegar (2.4-3.4) |
| 3 | wine (3.5-3.7) |
| 4 | Orange juice. Apple juice (3.8). Beer. Tomatoes. |
| 5 | Cottage cheese. Black coffee. Rain water 5.6. |
| 6 | Milk. Fish (6.7-7). chicken (6.4-6.6). |
| 7 | Neutral: equal numbers of hydrogen and hydroxyl ions. Blood (7.3-7.4). Distilled water without CO2, after boiling. |
| 8 | Sea water (8.1). Egg white. |
| 9 | Borax. baking soda. |
| 10 | Milk of magnesia. Magnesium hydroxide Mg (OH)2. |
| 11 | Household ammonia |
| 12 | Photographic developer, household bleach |
| 13 | Oven cleaner |
| 14 | Sodium lye NaOH, 1 mol/litre. |


©2017 by e-Gizmo Mechatronix Central

# LAMPIRAN E
## Datasheet Sensor Suhu DS18B20

*Click here for production status of specific part numbers.*

**DS18B20**

**Programmable Resolution 1-Wire Digital Thermometer**

### General Description

The DS18B20 digital thermometer provides 9-bit to 12-bit Celsius temperature measurements and has an alarm function with nonvolatile user-programmable upper and lower trigger points. The DS18B20 communicates over a 1-Wire bus that by definition requires only one data line (and ground) for communication with a central microprocessor. In addition, the DS18B20 can derive power directly from the data line ("parasite power"), eliminating the need for an external power supply.

Each DS18B20 has a unique 64-bit serial code, which allows multiple DS18B20s to function on the same 1-Wire bus. Thus, it is simple to use one microprocessor to control many DS18B20s distributed over a large area. Applications that can benefit from this feature include HVAC environmental controls, temperature monitoring systems inside buildings, equipment, or machinery, and process monitoring and control systems.

### Applications

- Thermostatic Controls
- Industrial Systems
- Consumer Products
- Thermometers
- Thermally Sensitive Systems

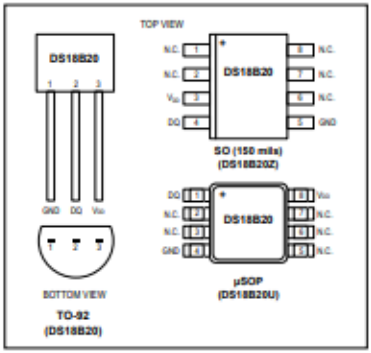
*Ordering Information* ***appears at end of data sheet.***

*1-Wire is a registered trademark of Maxim Integrated Products, Inc.*

### Benefits and Features

- Unique 1-Wire® Interface Requires Only One Port Pin for Communication
- Reduce Component Count with Integrated Temperature Sensor and EEPROM
  - Measures Temperatures from -55°C to +125°C (-67°F to +257°F)
  - ±0.5°C Accuracy from -10°C to +85°C
  - Programmable Resolution from 9 Bits to 12 Bits
  - No External Components Required
- Parasitic Power Mode Requires Only 2 Pins for Operation (DQ and GND)
- Simplifies Distributed Temperature-Sensing Applications with Multidrop Capability
  - Each Device Has a Unique 64-Bit Serial Code Stored in On-Board ROM
- Flexible User-Definable Nonvolatile (NV) Alarm Settings with Alarm Search Command Identifies Devices with Temperatures Outside Programmed Limits
- Available in 8-Pin SO (150 mils), 8-Pin µSOP, and 3-Pin TO-92 Packages

### Pin Configurations




maxim integrated.

19-7487; Rev 6; 7/19

## Absolute Maximum Ratings

Voltage Range on Any Pin Relative to Ground ..... -0.5V to +6.0V
Operating Temperature Range ..... -55°C to +125°C
Storage Temperature Range ..... -55°C to +125°C
Solder Temperature ..... Refer to the IPC/JEDEC J-STD-020 Specification.

*These are stress ratings only and functional operation of the device at these or any other conditions above those indicated in the operation sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods of time may affect reliability.*

## DC Electrical Characteristics

(-55°C to +125°C; $V_{DD}$ = 3.0V to 5.5V)

| PARAMETER | SYMBOL | CONDITIONS | | MIN | TYP | MAX | UNITS |
|---|---|---|---|---|---|---|---|
| Supply Voltage | $V_{DD}$ | Local power (Note 1) | | +3.0 | | +5.5 | V |
| Pullup Supply Voltage | $V_{PU}$ | Parasite power | (Notes 1, 2) | +3.0 | | +5.5 | V |
| | | Local power | | +3.0 | | $V_{DD}$ | |
| Thermometer Error | $t_{ERR}$ | -10°C to +85°C | (Note 3) | | | ±0.5 | °C |
| | | -30°C to +100°C | | | | ±1 | |
| | | -55°C to +125°C | | | | ±2 | |
| Input Logic-Low | $V_{IL}$ | (Notes 1, 4, 5) | | -0.3 | | +0.8 | V |
| Input Logic-High | $V_{IH}$ | Local power | (Notes 1,6) | +2.2 | | The lower of 5.5 or $V_{DD}$ + 0.3 | V |
| | | Parasite power | | +3.0 | | | |
| Sink Current | $I_L$ | $V_{I/O}$ = 0.4V | | 4.0 | | | mA |
| Standby Current | $I_{DDS}$ | (Notes 7, 8) | | | 750 | 1000 | nA |
| Active Current | $I_{DD}$ | $V_{DD}$ = 5V (Note 9) | | | 1 | 1.5 | mA |
| DQ Input Current | $I_{DQ}$ | (Note 10) | | | 5 | | µA |
| Drift | | (Note 11) | | | ±0.2 | | °C |

**Note 1:** All voltages are referenced to ground.
**Note 2:** The Pullup Supply Voltage specification assumes that the pullup device is ideal, and therefore the high level of the pullup is equal to $V_{PU}$. In order to meet the $V_{IH}$ spec of the DS18B20, the actual supply rail for the strong pullup transistor must include margin for the voltage drop across the transistor when it is turned on; thus: $V_{PU_ACTUAL} = V_{PU_IDEAL} + V_{TRANSISTOR}$.
**Note 3:** See typical performance curve in Figure 1. Thermometer Error limits are 3-sigma values.
**Note 4:** Logic-low voltages are specified at a sink current of 4mA.
**Note 5:** To guarantee a presence pulse under low voltage parasite power conditions, $V_{ILMAX}$ may have to be reduced to as low as 0.5V.
**Note 6:** Logic-high voltages are specified at a source current of 1mA.
**Note 7:** Standby current specified up to +70°C. Standby current typically is 3µA at +125°C.
**Note 8:** To minimize $I_{DDS}$, DQ should be within the following ranges: GND ≤ DQ ≤ GND + 0.3V or $V_{DD}$ – 0.3V ≤ DQ ≤ $V_{DD}$.
**Note 9:** Active current refers to supply current during active temperature conversions or EEPROM writes.
**Note 10:** DQ line is high ("high-Z" state).
**Note 11:** Drift data is based on a 1000-hour stress test at +125°C with $V_{DD}$ = 5.5V.

## AC Electrical Characteristics–NV Memory

(-55°C to +125°C; $V_{DD}$ = 3.0V to 5.5V)

| PARAMETER | SYMBOL | CONDITIONS | MIN | TYP | MAX | UNITS |
|---|---|---|---|---|---|---|
| NV Write Cycle Time | $t_{WR}$ | | | 2 | 10 | ms |
| EEPROM Writes | $N_{EEWR}$ | -55°C to +55°C | 50k | | | writes |
| EEPROM Data Retention | $t_{EEDR}$ | -55°C to +55°C | 10 | | | years |

## AC Electrical Characteristics

(-55°C to +125°C; $V_{DD}$ = 3.0V to 5.5V)

| PARAMETER | SYMBOL | CONDITIONS | | MIN | TYP | MAX | UNITS |
|---|---|---|---|---|---|---|---|
| Temperature Conversion Time | $t_{CONV}$ | 9-bit resolution | (Note 12) | | | 93.75 | ms |
| | | 10-bit resolution | | | | 187.5 | |
| | | 11-bit resolution | | | | 375 | |
| | | 12-bit resolution | | | | 750 | |
| Time to Strong Pullup On | $t_{SPON}$ | Start convert T command issued | | | | 10 | µs |
| Time Slot | $t_{SLOT}$ | (Note 12) | | 60 | | 120 | µs |
| Recovery Time | $t_{REC}$ | (Note 12) | | 1 | | | µs |
| Write 0 Low Time | $t_{LOW0}$ | (Note 12) | | 60 | | 120 | µs |
| Write 1 Low Time | $t_{LOW1}$ | (Note 12) | | 1 | | 15 | µs |
| Read Data Valid | $t_{RDV}$ | (Note 12) | | | | 15 | µs |
| Reset Time High | $t_{RSTH}$ | (Note 12) | | 480 | | | µs |
| Reset Time Low | $t_{RSTL}$ | (Notes 12, 13) | | 480 | | | µs |
| Presence-Detect High | $t_{PDHIGH}$ | (Note 12) | | 15 | | 60 | µs |
| Presence-Detect Low | $t_{PDLOW}$ | (Note 12) | | 60 | | 240 | µs |
| Capacitance | $C_{IN/OUT}$ | | | | | 25 | pF |

**Note 12:** See the timing diagrams in Figure 2.
**Note 13:** Under parasite power, if $t_{RSTL}$ > 960µs, a power-on reset can occur.

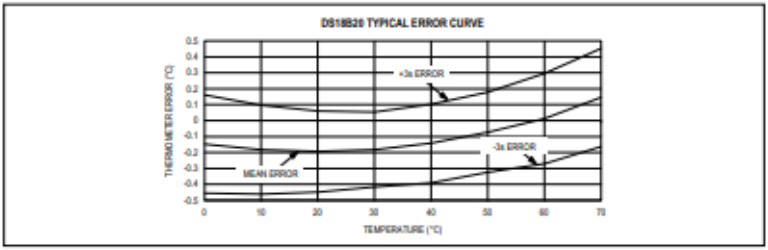



*Figure 1. Typical Performance Curve*

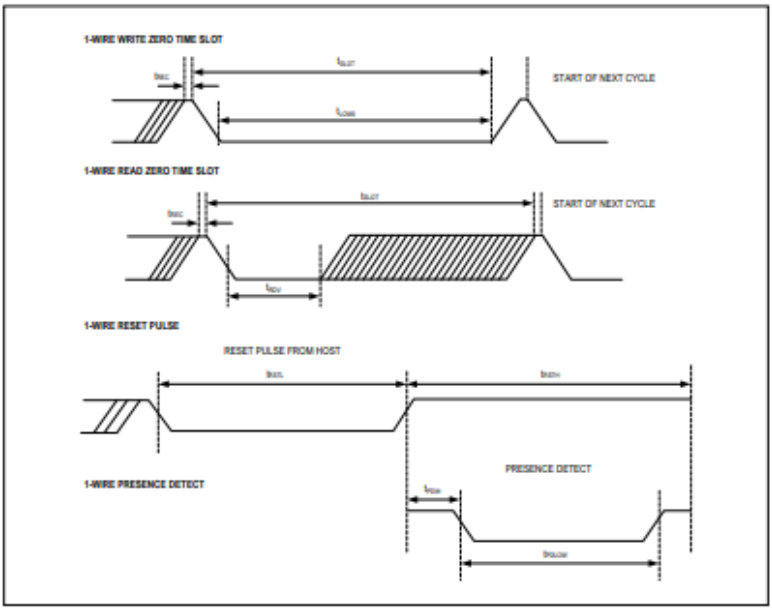



*Figure 2. Timing Diagrams*

## Pin Description

| PIN | | | NAME | FUNCTION |
|---|---|---|---|---|
| SO | µSOP | TO-92 | | |
| 1, 2, 6, 7, 8 | 2, 3, 5, 6, 7 | — | N.C. | No Connection |
| 3 | 8 | 3 | $V_{DD}$ | Optional $V_{DD}$. $V_{DD}$ must be grounded for operation in parasite power mode. |
| 4 | 1 | 2 | DQ | Data Input/Output. Open-drain 1-Wire interface pin. Also provides power to the device when used in parasite power mode (see the *Powering the DS18B20* section.) |
| 5 | 4 | 1 | GND | Ground |

# LAMPIRAN F
## Datasheet IRF540N

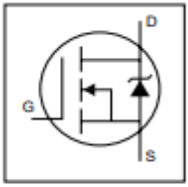
International
IOR Rectifier

PD - 91341B

# IRF540N

HEXFET® Power MOSFET

- Advanced Process Technology
- Ultra Low On-Resistance
- Dynamic dv/dt Rating
- 175°C Operating Temperature
- Fast Switching
- Fully Avalanche Rated




$V_{DSS}$ = 100V

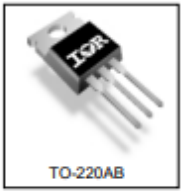
$R_{DS(on)}$ = 44mΩ

$I_D$ = 33A

### Description

Advanced HEXFET® Power MOSFETs from International Rectifier utilize advanced processing techniques to achieve extremely low on-resistance per silicon area. This benefit, combined with the fast switching speed and ruggedized device design that HEXFET power MOSFETs are well known for, provides the designer with an extremely efficient and reliable device for use in a wide variety of applications.

The TO-220 package is universally preferred for all commercial-industrial applications at power dissipation levels to approximately 50 watts. The low thermal resistance and low package cost of the TO-220 contribute to its wide acceptance throughout the industry.



TO-220AB

### Absolute Maximum Ratings

| | Parameter | Max. | Units |
|---|---|---|---|
| $I_D$ @ $T_C$ = 25°C | Continuous Drain Current, $V_{GS}$ @ 10V | 33 | A |
| $I_D$ @ $T_C$ = 100°C | Continuous Drain Current, $V_{GS}$ @ 10V | 23 | |
| $I_{DM}$ | Pulsed Drain Current ① | 110 | |
| $P_D$ @$T_C$ = 25°C | Power Dissipation | 130 | W |
| | Linear Derating Factor | 0.87 | W/°C |
| $V_{GS}$ | Gate-to-Source Voltage | ± 20 | V |
| $I_{AR}$ | Avalanche Current① | 16 | A |
| $E_{AR}$ | Repetitive Avalanche Energy① | 13 | mJ |
| dv/dt | Peak Diode Recovery dv/dt ③ | 7.0 | V/ns |
| $T_J$<br>$T_{STG}$ | Operating Junction and<br>Storage Temperature Range | -55 to + 175 | °C |
| | Soldering Temperature, for 10 seconds | 300 (1.6mm from case ) | |
| | Mounting torque, 6-32 or M3 srew | 10 lbf•in (1.1N•m) | |

### Thermal Resistance

| | Parameter | Typ. | Max. | Units |
|---|---|---|---|---|
| $R_{\theta JC}$ | Junction-to-Case | — | 1.15 | °C/W |
| $R_{\theta CS}$ | Case-to-Sink, Flat, Greased Surface | 0.50 | — | |
| $R_{\theta JA}$ | Junction-to-Ambient | — | 62 | |

## Electrical Characteristics @ $T_J$ = 25°C (unless otherwise specified)

| | Parameter | Min. | Typ. | Max. | Units | Conditions |
|---|---|---|---|---|---|---|
| $V_{(BR)DSS}$ | Drain-to-Source Breakdown Voltage | 100 | — | — | V | $V_{GS}$ = 0V, $I_D$ = 250µA |
| $\Delta V_{(BR)DSS}/\Delta T_J$ | Breakdown Voltage Temp. Coefficient | — | 0.12 | — | V/°C | Reference to 25°C, $I_D$ = 1mA |
| $R_{DS(on)}$ | Static Drain-to-Source On-Resistance | — | — | 44 | mΩ | $V_{GS}$ = 10V, $I_D$ = 16A ④ |
| $V_{GS(th)}$ | Gate Threshold Voltage | 2.0 | — | 4.0 | V | $V_{DS}$ = $V_{GS}$, $I_D$ = 250µA |
| $g_{fs}$ | Forward Transconductance | 21 | — | — | S | $V_{DS}$ = 50V, $I_D$ = 16A④ |
| $I_{DSS}$ | Drain-to-Source Leakage Current | — | — | 25 | µA | $V_{DS}$ = 100V, $V_{GS}$ = 0V |
| | | — | — | 250 | | $V_{DS}$ = 80V, $V_{GS}$ = 0V, $T_J$ = 150°C |
| $I_{GSS}$ | Gate-to-Source Forward Leakage | — | — | 100 | nA | $V_{GS}$ = 20V |
| | Gate-to-Source Reverse Leakage | — | — | -100 | | $V_{GS}$ = -20V |
| $Q_g$ | Total Gate Charge | — | — | 71 | nC | $I_D$ = 16A |
| $Q_{gs}$ | Gate-to-Source Charge | — | — | 14 | | $V_{DS}$ = 80V |
| $Q_{gd}$ | Gate-to-Drain ("Miller") Charge | — | — | 21 | | $V_{GS}$ = 10V, See Fig. 6 and 13 |
| $t_{d(on)}$ | Turn-On Delay Time | — | 11 | — | ns | $V_{DD}$ = 50V |
| $t_r$ | Rise Time | — | 35 | — | | $I_D$ = 16A |
| $t_{d(off)}$ | Turn-Off Delay Time | — | 39 | — | | $R_G$ = 5.1Ω |
| $t_f$ | Fall Time | — | 35 | — | | $V_{GS}$ = 10V, See Fig. 10 ④ |
| $L_D$ | Internal Drain Inductance | — | 4.5 | — | nH | Between lead, 6mm (0.25in.) from package and center of die contact |
| $L_S$ | Internal Source Inductance | — | 7.5 | — | | |
| $C_{iss}$ | Input Capacitance | — | 1960 | — | pF | $V_{GS}$ = 0V |
| $C_{oss}$ | Output Capacitance | — | 250 | — | | $V_{DS}$ = 25V |
| $C_{rss}$ | Reverse Transfer Capacitance | — | 40 | — | | $f$ = 1.0MHz, See Fig. 5 |
| $E_{AS}$ | Single Pulse Avalanche Energy② | — | 700⑤ | 185⑥ | mJ | $I_{AS}$ = 16A, L = 1.5mH |

## Source-Drain Ratings and Characteristics

| | Parameter | Min. | Typ. | Max. | Units | Conditions |
|---|---|---|---|---|---|---|
| $I_S$ | Continuous Source Current (Body Diode) | — | — | 33 | A | MOSFET symbol showing the integral reverse p-n junction diode. |
| $I_{SM}$ | Pulsed Source Current (Body Diode)① | — | — | 110 | | |
| $V_{SD}$ | Diode Forward Voltage | — | — | 1.2 | V | $T_J$ = 25°C, $I_S$ = 16A, $V_{GS}$ = 0V ④ |
| $t_{rr}$ | Reverse Recovery Time | — | 115 | 170 | ns | $T_J$ = 25°C, $I_F$ = 16A |
| $Q_{rr}$ | Reverse Recovery Charge | — | 505 | 760 | nC | di/dt = 100A/µs ④ |
| $t_{on}$ | Forward Turn-On Time | Intrinsic turn-on time is negligible (turn-on is dominated by $L_S$+$L_D$) | | | | |

**Notes:**

① Repetitive rating; pulse width limited by max. junction temperature. (See fig. 11)

② Starting $T_J$ = 25°C, L = 1.5mH
$R_G$ = 25Ω, $I_{AS}$ = 16A. (See Figure 12)

③ $I_{SD}$ ≤ 16A, di/dt ≤ 340A/µs, $V_{DD}$ ≤ $V_{(BR)DSS}$, $T_J$ ≤ 175°C

④ Pulse width ≤ 400µs; duty cycle ≤ 2%.

⑤ This is a typical value at device destruction and represents operation outside rated limits.

⑥ This is a calculated value limited to $T_J$ = 175°C .

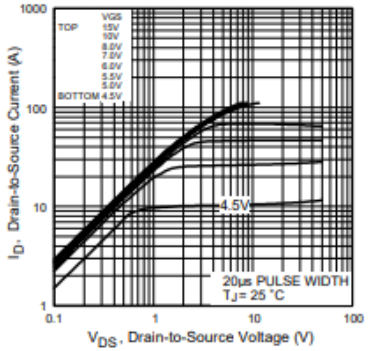



**Fig 1.** Typical Output Characteristics

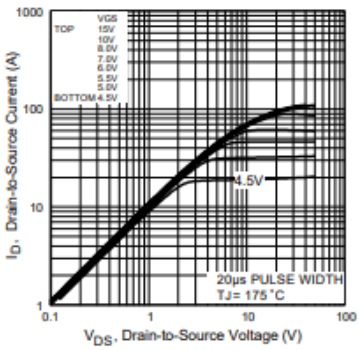



**Fig 2.** Typical Output Characteristics

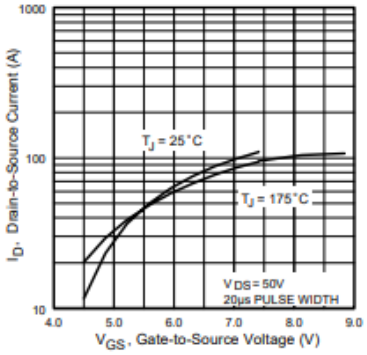



**Fig 3.** Typical Transfer Characteristics

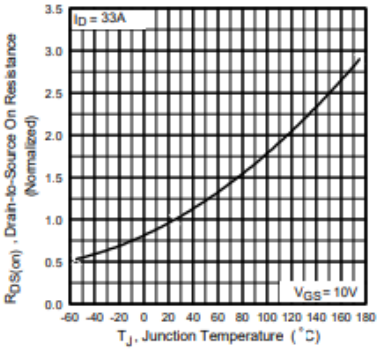



**Fig 4.** Normalized On-Resistance Vs. Temperature

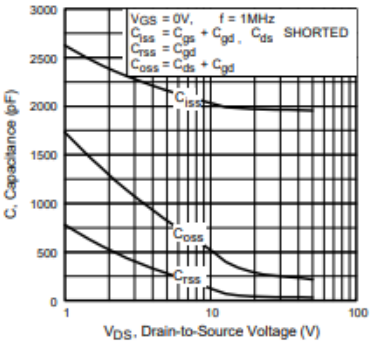



**Fig 5.** Typical Capacitance Vs. Drain-to-Source Voltage

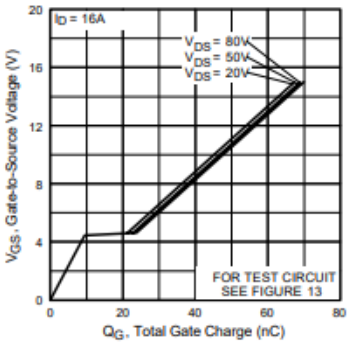



**Fig 6.** Typical Gate Charge Vs. Gate-to-Source Voltage

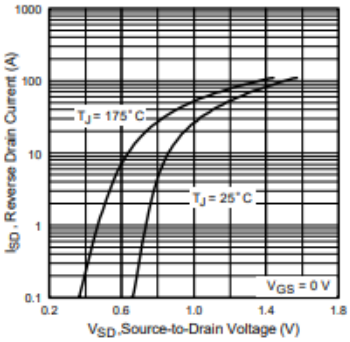



**Fig 7.** Typical Source-Drain Diode Forward Voltage

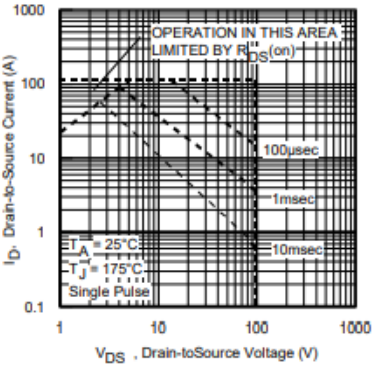



**Fig 8.** Maximum Safe Operating Area

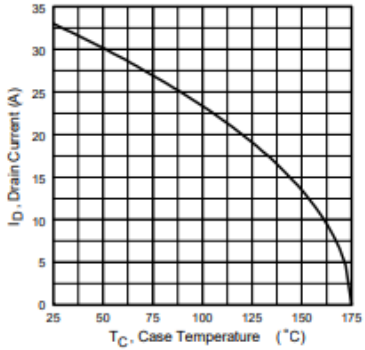



**Fig 9.** Maximum Drain Current Vs. Case Temperature

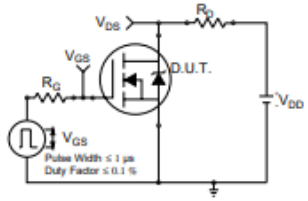



**Fig 10a.** Switching Time Test Circuit

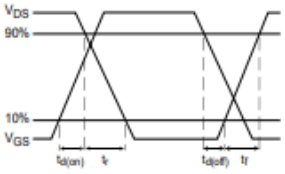



**Fig 10b.** Switching Time Waveforms

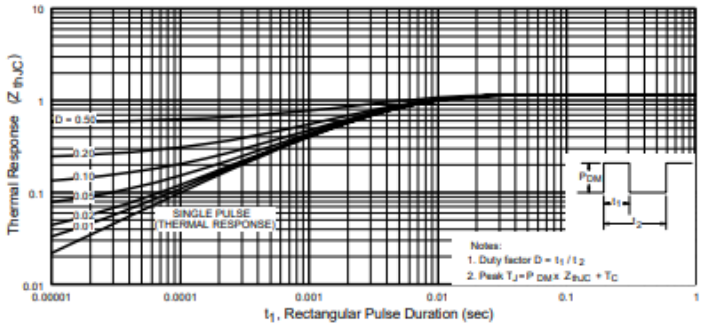



**Fig 11.** Maximum Effective Transient Thermal Impedance, Junction-to-Case

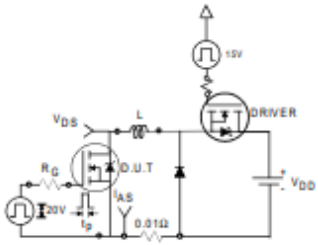



**Fig 12a.** Unclamped Inductive Test Circuit

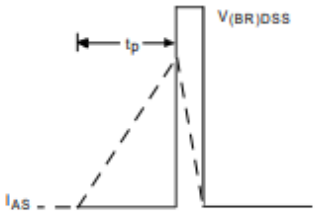



**Fig 12b.** Unclamped Inductive Waveforms

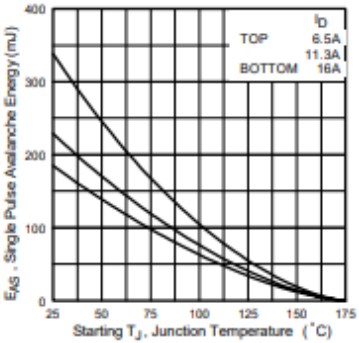



**Fig 12c.** Maximum Avalanche Energy Vs. Drain Current

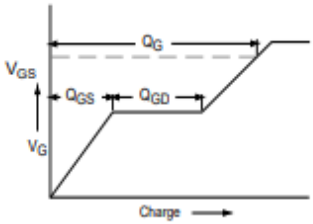



**Fig 13a.** Basic Gate Charge Waveform

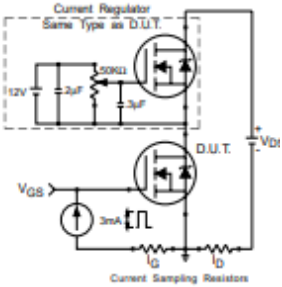
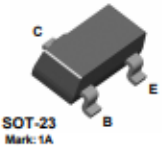


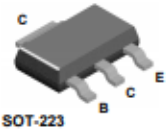

**Fig 13b.** Gate Charge Test Circuit

# LAMPIRAN G
## Datasheet 2N3904

2N3904/MMBT3904/PZT3904

**2N3904** **MMBT3904** **PZT3904**




## NPN General Purpose Amplifier

This device is designed as a general purpose amplifier and switch. The useful dynamic range extends to 100 mA as a switch and to 100 MHz as an amplifier.

### Absolute Maximum Ratings*

$T_A$ = 25°C unless otherwise noted

| Symbol | Parameter | Value | Units |
|---|---|---|---|
| $V_{CEO}$ | Collector-Emitter Voltage | 40 | V |
| $V_{CBO}$ | Collector-Base Voltage | 60 | V |
| $V_{EBO}$ | Emitter-Base Voltage | 6.0 | V |
| $I_C$ | Collector Current - Continuous | 200 | mA |
| $T_J$, $T_{stg}$ | Operating and Storage Junction Temperature Range | -55 to +150 | °C |

*These ratings are limiting values above which the serviceability of any semiconductor device may be impaired.

**NOTES:**
1) These ratings are based on a maximum junction temperature of 150 degrees C.
2) These are steady state limits. The factory should be consulted on applications involving pulsed or low duty cycle operations.

### Thermal Characteristics

$T_A$ = 25°C unless otherwise noted

| Symbol | Characteristic | Max | | | Units |
|---|---|---|---|---|---|
| | | 2N3904 | *MMBT3904 | **PZT3904 | |
| $P_D$ | Total Device Dissipation<br>Derate above 25°C | 625<br>5.0 | 350<br>2.8 | 1,000<br>8.0 | mW<br>mW/°C |
| $R_{\theta JC}$ | Thermal Resistance, Junction to Case | 83.3 | | | °C/W |
| $R_{\theta JA}$ | Thermal Resistance, Junction to Ambient | 200 | 357 | 125 | °C/W |

*Device mounted on FR-4 PCB 1.6" X 1.6" X 0.06."

**Device mounted on FR-4 PCB 36 mm X 18 mm X 1.5 mm; mounting pad for the collector lead min. 6 $cm^2$.

**NPN General Purpose Amplifier**
(continued)

## Electrical Characteristics

$T_A$ = 25°C unless otherwise noted

| Symbol | Parameter | Test Conditions | Min | Max | Units |
|---|---|---|---|---|---|
| **OFF CHARACTERISTICS** | | | | | |
| $V_{(BR)CEO}$ | Collector-Emitter Breakdown Voltage | $I_C$ = 1.0 mA, $I_B$ = 0 | 40 | | V |
| $V_{(BR)CBO}$ | Collector-Base Breakdown Voltage | $I_C$ = 10 µA, $I_E$ = 0 | 60 | | V |
| $V_{(BR)EBO}$ | Emitter-Base Breakdown Voltage | $I_E$ = 10 µA, $I_C$ = 0 | 6.0 | | V |
| $I_{BL}$ | Base Cutoff Current | $V_{CE}$ = 30 V, $V_{EB}$ = 3V | | 50 | nA |
| $I_{CEX}$ | Collector Cutoff Current | $V_{CE}$ = 30 V, $V_{EB}$ = 3V | | 50 | nA |
| **ON CHARACTERISTICS*** | | | | | |
| $h_{FE}$ | DC Current Gain | $I_C$ = 0.1 mA, $V_{CE}$ = 1.0 V<br>$I_C$ = 1.0 mA, $V_{CE}$ = 1.0 V<br>$I_C$ = 10 mA, $V_{CE}$ = 1.0 V<br>$I_C$ = 50 mA, $V_{CE}$ = 1.0 V<br>$I_C$ = 100 mA, $V_{CE}$ = 1.0 V | 40<br>70<br>100<br>60<br>30 | <br><br>300<br><br> | |
| $V_{CE(sat)}$ | Collector-Emitter Saturation Voltage | $I_C$ = 10 mA, $I_B$ = 1.0 mA<br>$I_C$ = 50 mA, $I_B$ = 5.0 mA | | 0.2<br>0.3 | V<br>V |
| $V_{BE(sat)}$ | Base-Emitter Saturation Voltage | $I_C$ = 10 mA, $I_B$ = 1.0 mA<br>$I_C$ = 50 mA, $I_B$ = 5.0 mA | 0.65 | 0.85<br>0.95 | V<br>V |
| **SMALL SIGNAL CHARACTERISTICS** | | | | | |
| $f_T$ | Current Gain - Bandwidth Product | $I_C$ = 10 mA, $V_{CE}$ = 20 V, f = 100 MHz | 300 | | MHz |
| $C_{obo}$ | Output Capacitance | $V_{CB}$ = 5.0 V, $I_E$ = 0, f = 1.0 MHz | | 4.0 | pF |
| $C_{ibo}$ | Input Capacitance | $V_{EB}$ = 0.5 V, $I_C$ = 0, f = 1.0 MHz | | 8.0 | pF |
| NF | Noise Figure | $I_C$ = 100 µA, $V_{CE}$ = 5.0 V, $R_S$ = 1.0kΩ, f=10 Hz to 15.7kHz | | 5.0 | dB |
| **SWITCHING CHARACTERISTICS** | | | | | |
| $t_d$ | Delay Time | $V_{CC}$ = 3.0 V, $V_{BE}$ = 0.5 V, | | 35 | ns |
| $t_r$ | Rise Time | $I_C$ = 10 mA, $I_{B1}$ = 1.0 mA | | 35 | ns |
| $t_s$ | Storage Time | $V_{CC}$ = 3.0 V, $I_C$ = 10mA | | 200 | ns |
| $t_f$ | Fall Time | $I_{B1}$ = $I_{B2}$ = 1.0 mA | | 50 | ns |

*Pulse Test: Pulse Width ≤300 µs, Duty Cycle ≤2.0%

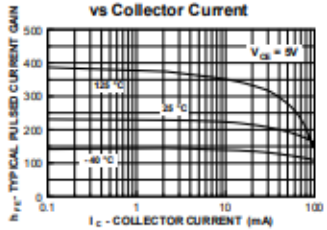
## Spice Model

NPN (Is=6.734f Xti=3 Eg=1.11 Vaf=74.03 Bf=416.4 Ne=1.259 Ise=6.734 Ikf=66.78m Xtb=1.5 Br=.7371 Nc=2 Isc=0 Ikr=0 Rc=1 Cjc=3.638p Mjc=.3085 Vjc=.75 Fc=.5 Cje=4.493p Mje=.2593 Vje=.75 Tr=239.5n Tf=301.2p Itf=.4 Vtf=4 Xtf=2 Rb=10)

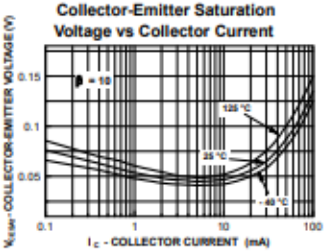
## NPN General Purpose Amplifier
(continued)

### Typical Characteristics

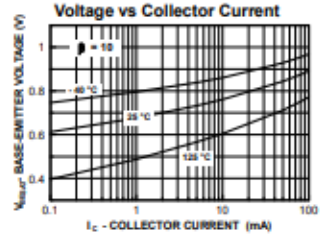
**Typical Pulsed Current Gain vs Collector Current**



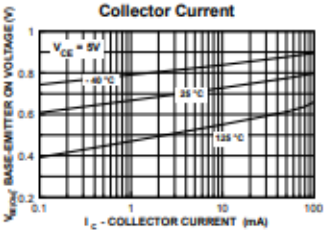

**Collector-Emitter Saturation Voltage vs Collector Current**



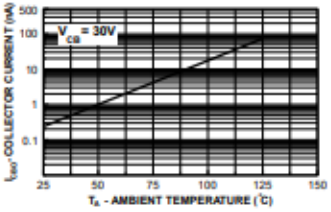

**Base-Emitter Saturation Voltage vs Collector Current**



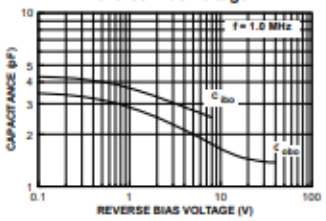

**Base-Emitter ON Voltage vs Collector Current**




**Collector-Cutoff Current vs Ambient Temperature**



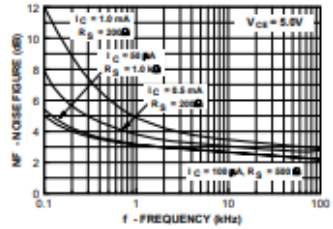

**Capacitance vs Reverse Bias Voltage**



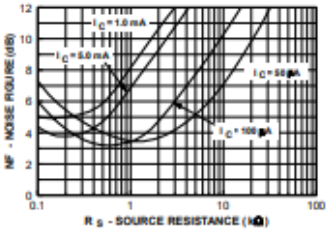
**NPN General Purpose Amplifier**
(continued)

## Typical Characteristics (continued)

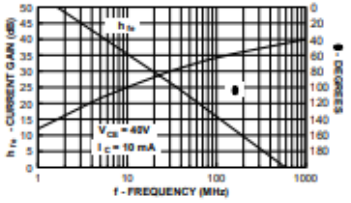
### Noise Figure vs Frequency



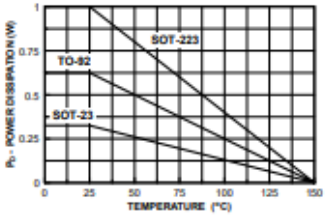

### Noise Figure vs Source Resistance



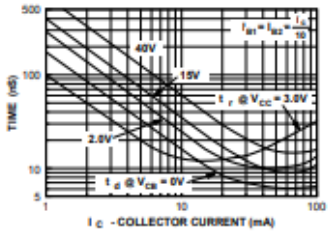

### Current Gain and Phase Angle vs Frequency



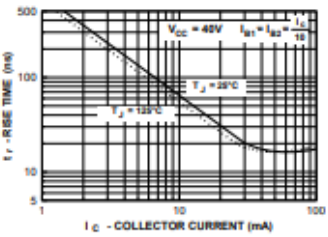

### Power Dissipation vs Ambient Temperature




### Turn-On Time vs Collector Current



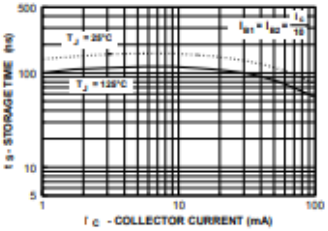

### Rise Time vs Collector Current



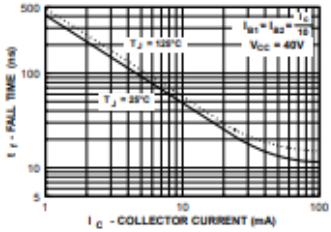
**NPN General Purpose Amplifier**
(continued)

## Typical Characteristics (continued)

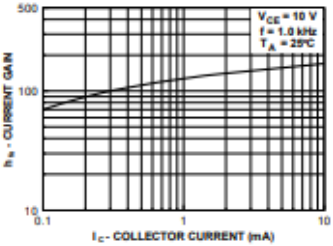
Storage Time vs Collector Current



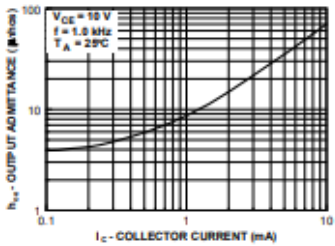

Fall Time vs Collector Current



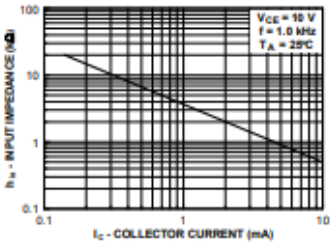

Current Gain



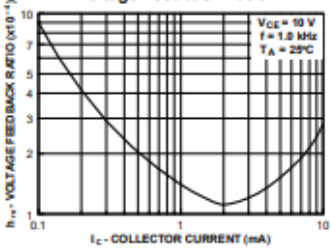

Output Admittance




Input Impedance




Voltage Feedback Ratio



## Test Circuits

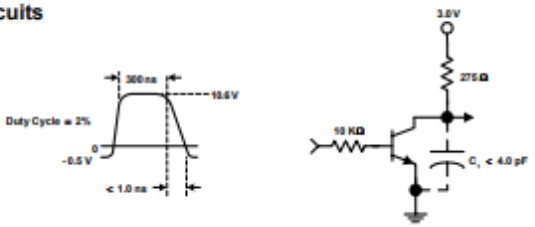



**FIGURE 1: Delay and Rise Time Equivalent Test Circuit**

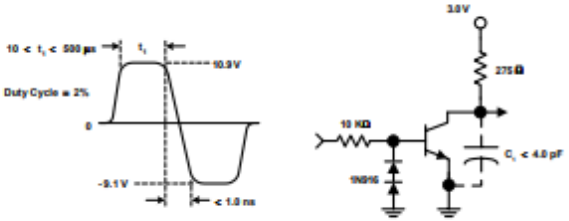



**FIGURE 2: Storage and Fall Time Equivalent Test Circuit**

# LAMPIRAN H
## Datasheet RTC DS3231

Rev 2; 6/05

DALLAS SEMICONDUCTOR MAXIM

# Extremely Accurate I²C-Integrated RTC/TCXO/Crystal

## General Description

The DS3231 is a low-cost, extremely accurate I²C real-time clock (RTC) with an integrated temperature-compensated crystal oscillator (TCXO) and crystal. The device incorporates a battery input, and maintains accurate timekeeping when main power to the device is interrupted. The integration of the crystal resonator enhances the long-term accuracy of the device as well as reduces the piece-part count in a manufacturing line. The DS3231 is available in commercial and industrial temperature ranges, and is offered in a 16-pin, 300-mil SO package.

The RTC maintains seconds, minutes, hours, day, date, month, and year information. The date at the end of the month is automatically adjusted for months with fewer 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. Two programmable time-of-day alarms and a programmable square-wave output are provided. Address and data are transferred serially through an I²C bidirectional bus.

A precision temperature-compensated voltage reference and comparator circuit monitors the status of $V_{CC}$ to detect power failures, to provide a reset output, and to automatically switch to the backup supply when necessary. Additionally, the $\overline{RST}$ pin is monitored as a pushbutton input for generating a reset externally.

## Applications

| | |
|---|---|
| Servers | Utility Power Meters |
| Telematics | GPS |

**Pin Configuration appears at end of data sheet.**

## Features

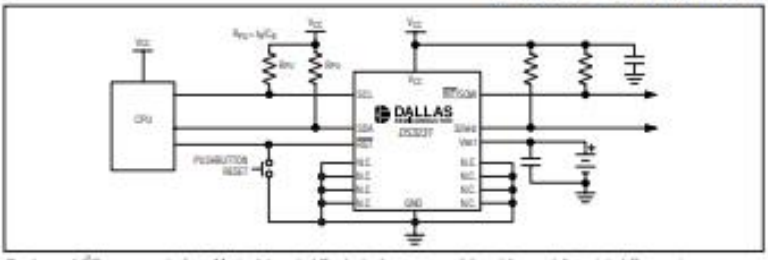
- Accuracy ±2ppm from 0°C to +40°C
- Accuracy ±3.5ppm from -40°C to +85°C
- Battery Backup Input for Continuous Timekeeping
- Operating Temperature Ranges
  Commercial: 0°C to +70°C
  Industrial: -40°C to +85°C
- Low-Power Consumption
- Real-Time Clock Counts Seconds, Minutes, Hours, Day, Date, Month, and Year with Leap Year Compensation Valid Up to 2100
- Two Time-of-Day Alarms
- Programmable Square-Wave Output
- Fast (400kHz) I²C Interface
- 3.3V Operation
- Digital Temp Sensor Output: ±3°C Accuracy
- Register for Aging Trim
- RST Input/Output
- UL Recognized

## Ordering Information

| PART | TEMP RANGE | PIN-PACKAGE | TOP MARK |
|---|---|---|---|
| DS3231S | 0°C to +70°C | 16 SO | DS3231 |
| DS3231SN | -40°C to +85°C | 16 SO | DS3231N |
| DS3231S+ | 0°C to +70°C | 16 SO | DS3231+ |
| DS3231SN+ | -40°C to +85°C | 16 SO | DS3231N+ |

+Denotes lead-free

## Typical Operating Circuit




*Purchase of I²C components from Maxim Integrated Products, Inc., or one of its sublicensed Associated Companies, conveys a license under the Philips I²C Patent Rights to use these components in an I²C system, provided that the system conforms to the I²C Standard Specification as defined by Philips.*

## ABSOLUTE MAXIMUM RATINGS

Voltage Range on $V_{CC}$, $V_{BAT}$, 32kHz, SCL, SDA, $\overline{RST}$, $\overline{INT}$/SQW Relative to Ground ..... -0.3V to +6.0V
Operating Temperature Range (noncondensing) ..... -40°C to +85°C
Junction Temperature ..... +125°C
Storage Temperature Range ..... -40°C to +85°C
Lead Temperature (Soldering, 10s) ..... +260°C/10s
Soldering Temperature ..... See the *Handling, PC Board Layout, and Assembly* section.

*Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

## RECOMMENDED DC OPERATING CONDITIONS

($T_A$ = $T_{MIN}$ to $T_{MAX}$, unless otherwise noted.) (Notes 1, 2)

| PARAMETER | SYMBOL | CONDITIONS | MIN | TYP | MAX | UNITS |
|---|---|---|---|---|---|---|
| Supply Voltage | $V_{CC}$ | | 2.3 | 3.3 | 5.5 | V |
| | $V_{BAT}$ | | 2.3 | 3.0 | 5.5 | V |
| Logic 1 Input SDA, SCL | $V_{IH}$ | | 0.7 x $V_{CC}$ | | $V_{CC}$ + 0.3 | V |
| Logic 0 Input SDA, SCL | $V_{IL}$ | | -0.3 | | +0.3 x $V_{CC}$ | V |
| Pullup Voltage (SDA, SCL, 32kHz, $\overline{INT}$/SQW) | $V_{PU}$ | $V_{CC}$ = 0V | | | 5.5V | V |

## ELECTRICAL CHARACTERISTICS

($V_{CC}$ = 2.3V to 5.5V, $V_{CC}$ > $V_{BAT}$, $T_A$ = $T_{MIN}$ to $T_{MAX}$, unless otherwise noted.) (Typical values are at $V_{CC}$ = 3.3V, $V_{BAT}$ = 3.0V, and $T_A$ = +25°C, unless otherwise noted.) (Notes 1, 2)

| PARAMETER | SYMBOL | CONDITIONS | | MIN | TYP | MAX | UNITS |
|---|---|---|---|---|---|---|---|
| Active Supply Current | $I_{CCA}$ | (Notes 3, 4) | $V_{CC}$ = 3.63V | | | 200 | µA |
| | | | $V_{CC}$ = 5.5V | | | 300 | |
| Standby Supply Current | $I_{CCS}$ | I²C bus inactive, 32kHz output on, SQW output off (Note 4) | $V_{CC}$ = 3.63V | | | 110 | µA |
| | | | $V_{CC}$ = 5.5V | | | 170 | |
| Temperature Conversion Current | $I_{CCSCONV}$ | I²C bus inactive, 32kHz output on, SQW output off | $V_{CC}$ = 3.63V | | | 575 | µA |
| | | | $V_{CC}$ = 5.5V | | | 650 | |
| Power-Fail Voltage | $V_{PF}$ | | | 2.45 | 2.575 | 2.70 | V |
| Logic 0 Output, 32kHz, $\overline{INT}$/SQW, SDA | $V_{OL}$ | $I_{OL}$ = 3mA | | | | 0.4 | V |
| Logic 0 Output, $\overline{RST}$ | $V_{OL}$ | $I_{OL}$ = 1mA | | | | 0.4 | V |
| Output Leakage Current 32kHz, $\overline{INT}$/SQW, SDA | $I_{LO}$ | Output high impedance | | -1 | 0 | +1 | µA |
| Input Leakage SCL | $I_{LI}$ | | | -1 | | +1 | µA |
| $\overline{RST}$ Pin I/O Leakage | $I_{OL}$ | $\overline{RST}$ high impedance (Note 5) | | -200 | | +10 | µA |
| $V_{BAT}$ Leakage Current ($V_{CC}$ Active) | $I_{BATLKG}$ | | | | 25 | 100 | nA |

## ELECTRICAL CHARACTERISTICS (continued)

(VCC = 2.3V to 5.5V, VCC > VBAT, TA = TMIN to TMAX, unless otherwise noted.) (Typical values are at VCC = 3.3V, VBAT = 3.0V, and TA = +25°C, unless otherwise noted.) (Notes 1, 2)

| PARAMETER | SYMBOL | CONDITIONS | | MIN | TYP | MAX | UNITS |
|---|---|---|---|---|---|---|---|
| Output Frequency | $f_{OUT}$ | VCC = 3.3V or VBAT = 3.3V | | | 32.768 | | kHz |
| Frequency Stability vs. Temperature (Commercial) | $\Delta f/f_{OUT}$ | VCC = 3.3V or VBAT = 3.3V, aging offset = 00h | 0°C to +40°C | | | ±2 | ppm |
| | | | >40°C to +70°C | | | ±3.5 | |
| Frequency Stability vs. Temperature (Industrial) | $\Delta f/f_{OUT}$ | VCC = 3.3V or VBAT = 3.3V, aging offset = 00h | -40°C to <0°C | | | ±3.5 | ppm |
| | | | 0°C to +40°C | | | ±2 | |
| | | | >40°C to +85°C | | | ±3.5 | |
| Frequency Stability vs. Voltage | Δf/V | | | | 1 | | ppm/V |
| Trim Register Frequency Sensitivity per LSB | Δf/LSB | Specified at: | -40°C | | 0.7 | | ppm |
| | | | +25°C | | 0.1 | | |
| | | | +70°C | | 0.4 | | |
| | | | +85°C | | 0.8 | | |
| Temperature Accuracy | Temp | VCC = 3.3V or VBAT = 3.3V | | -3 | | +3 | °C |
| Crystal Aging | $\Delta f/f_0$ | After reflow, not production tested | First year | | ±1.0 | | ppm |
| | | | 0–10 years | | ±5.0 | | |

## ELECTRICAL CHARACTERISTICS

**(VCC = 0V, VBAT = 2.3V to 5.5V**, TA = TMIN to TMAX, unless otherw.ise noted.) (Note 1)

| PARAMETER | SYMBOL | CONDITIONS | | MIN | TYP | MAX | UNITS |
|---|---|---|---|---|---|---|---|
| Active Battery Current | $I_{BATA}$ | $\overline{EOSC}$ = 0, BBSQW = 0, SCL = 400kHz (Note 4) | VBAT = 3.63V | | | 70 | µA |
| | | | VBAT = 5.5V | | | 150 | |
| Timekeeping Battery Current | $I_{BATT}$ | $\overline{EOSC}$ = 0, BBSQW = 0, EN32kHz = 1, SCL = SDA = 0V or SCL = SDA = VBAT (Note 4) | VBAT = 3.63V | | 0.84 | 3.0 | µA |
| | | | VBAT = 5.5V | | 1.0 | 3.5 | |
| Temperature Conversion Current | $I_{BATTC}$ | $\overline{EOSC}$ = 0, BBSQW = 0, SCL = SDA = 0V or SCL = SDA = VBAT | VBAT = 3.63V | | | 575 | µA |
| | | | VBAT = 5.5V | | | 650 | |
| Data-Retention Current | $I_{BATTDR}$ | $\overline{EOSC}$ = 1, SCL = SDA = 0V, +25°C | | | | 100 | nA |

## AC ELECTRICAL CHARACTERISTICS

($V_{CC}$ = $V_{CC(MIN)}$ to $V_{CC(MAX)}$ or $V_{BAT}$ = $V_{BAT(MIN)}$ to $V_{BAT(MAX)}$, $V_{BAT}$ > $V_{CC}$, $T_A$ = $T_{MIN}$ to $T_{MAX}$, unless otherwise noted.) (Note 1)

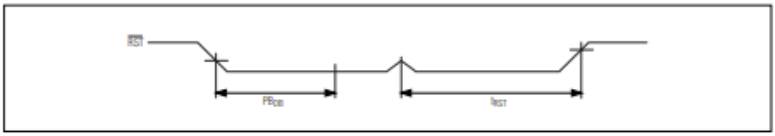
| PARAMETER | SYMBOL | CONDITIONS | MIN | TYP | MAX | UNITS |
|---|---|---|---|---|---|---|
| SCL Clock Frequency | $f_{SCL}$ | Fast mode | 100 | | 400 | kHz |
| | | Standard mode | 0 | | 100 | |
| Bus Free Time Between STOP and START Conditions | $t_{BUF}$ | Fast mode | 1.3 | | | µs |
| | | Standard mode | 4.7 | | | |
| Hold Time (Repeated) START Condition (Note 6) | $t_{HD:STA}$ | Fast mode | 0.6 | | | µs |
| | | Standard mode | 4.0 | | | |
| Low Period of SCL Clock | $t_{LOW}$ | Fast mode | 1.3 | | | µs |
| | | Standard mode | 4.7 | | | |
| High Period of SCL Clock | $t_{HIGH}$ | Fast mode | 0.6 | | | µs |
| | | Standard mode | 4.0 | | | |
| Data Hold Time (Notes 7, 8) | $t_{HD:DAT}$ | Fast mode | 0 | | 0.9 | µs |
| | | Standard mode | 0 | | 0.9 | |
| Data Setup Time (Note 9) | $t_{SU:DAT}$ | Fast mode | 100 | | | ns |
| | | Standard mode | 250 | | | |
| Start Setup Time | $t_{SU:STA}$ | Fast mode | 0.6 | | | µs |
| | | Standard mode | 4.7 | | | |
| Rise Time of Both SDA and SCL Signals (Note 10) | $t_R$ | Fast mode | 20 + 0.1$C_B$ | | 300 | ns |
| | | Standard mode | | | 1000 | |
| Fall Time of Both SDA and SCL Signals (Note 10) | $t_F$ | Fast mode | 20 + 0.1$C_B$ | | 300 | ns |
| | | Standard mode | | | 300 | |
| Setup Time for STOP Condition | $t_{SU:STO}$ | Fast mode | 0.6 | | | µs |
| | | Standard mode | 4.7 | | | |
| Capacitive Load for Each Bus Line (Note 10) | $C_B$ | | | | 400 | pF |
| Capacitance for SDA, SCL | $C_{I/O}$ | | | 10 | | pF |
| Pulse Width of Spikes That Must Be Suppressed by the Input Filter | $t_{SP}$ | | | 30 | | ns |
| Pushbutton Debounce | $PB_{DB}$ | | | 250 | | ms |
| Reset Active Time | $t_{RST}$ | | | 250 | | ms |
| Oscillator Stop Flag (OSF) Delay | $t_{OSF}$ | (Note 11) | | 100 | | ms |
| Temperature Conversion Time | $t_{CONV}$ | | | 125 | 200 | ms |

## POWER-SWITCH CHARACTERISTICS

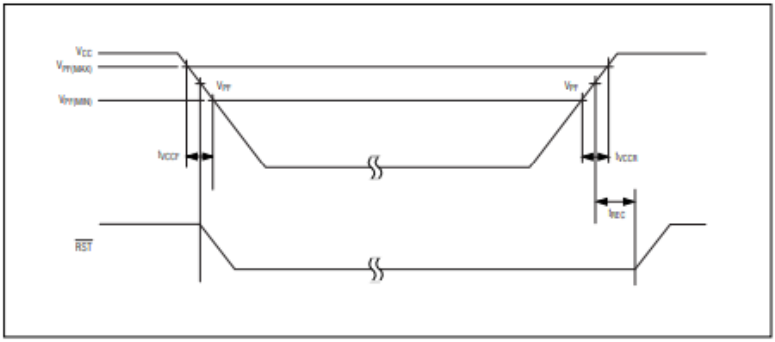
($T_A$ = $T_{MIN}$ to $T_{MAX}$)

| PARAMETER | SYMBOL | CONDITIONS | MIN | TYP | MAX | UNITS |
|---|---|---|---|---|---|---|
| $V_{CC}$ Fall Time; $V_{PF(MAX)}$ to $V_{PF(MIN)}$ | $t_{VCCF}$ | | 300 | | | µs |
| $V_{CC}$ Rise Time; $V_{PF(MIN)}$ to $V_{PF(MAX)}$ | $t_{VCCR}$ | | 0 | | | µs |
| Recovery at Power-Up | $t_{REC}$ | (Note 12) | | 250 | 300 | ms |

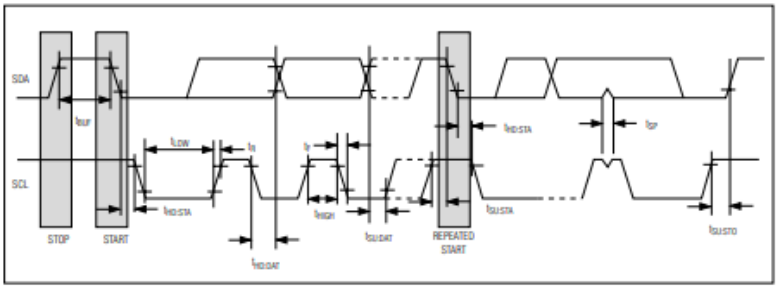
## Pushbutton Reset Timing




## Power-Switch Timing



**Data Transfer on I²C Serial Bus**

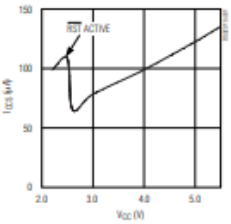



**Note 1:** Limits at -40°C are guaranteed by design and not production tested.
**Note 2:** All voltages are referenced to ground.
**Note 3:** $I_{CCA}$—SCL clocking at max frequency = 400kHz.
**Note 4:** Current is the averaged input current, which includes the temperature conversion current.
**Note 5:** The $\overline{RST}$ pin has an internal 50kΩ (nominal) pullup resistor to $V_{CC}$.
**Note 6:** After this period, the first clock pulse is generated.
**Note 7:** A device must internally provide a hold time of at least 300ns for the SDA signal (referred to the $V_{IH(MIN)}$ of the SCL signal) to bridge the undefined region of the falling edge of SCL.
**Note 8:** The maximum $t_{HD:DAT}$ needs only to be met if the device does not stretch the low period ($t_{LOW}$) of the SCL signal.
**Note 9:** A fast-mode device can be used in a standard-mode system, but the requirement $t_{SU:DAT} \geq 250ns$ must then be met. This is automatically the case if the device does not stretch the low period of the SCL signal. If such a device does stretch the low period of the SCL signal, it must output the next data bit to the SDA line $t_{R(MAX)} + t_{SU:DAT} = 1000 + 250 = 1250ns$ before the SCL line is released.
**Note 10:** $C_B$—total capacitance of one bus line in pF.
**Note 11:** The parameter $t_{OSF}$ is the period of time the oscillator must be stopped for the OSF flag to be set over the voltage range of $0.0V \leq V_{CC} \leq V_{CC(MAX)}$ and $2.3V \leq V_{BAT} \leq 3.4V$.
**Note 12:** This delay applies only if the oscillator is enabled and running. If the $\overline{EOSC}$ bit is a 1, the startup time of the oscillator is added to this delay.

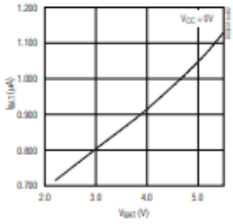
## Typical Operating Characteristics

($V_{CC}$ = +3.3V, $T_A$ = +25°C, unless otherwise noted.)

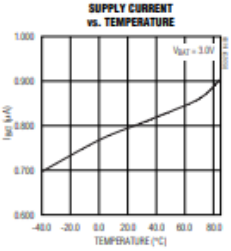
**STANDBY SUPPLY CURRENT vs. SUPPLY VOLTAGE**



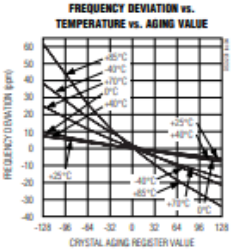

**SUPPLY CURRENT vs. SUPPLY VOLTAGE**



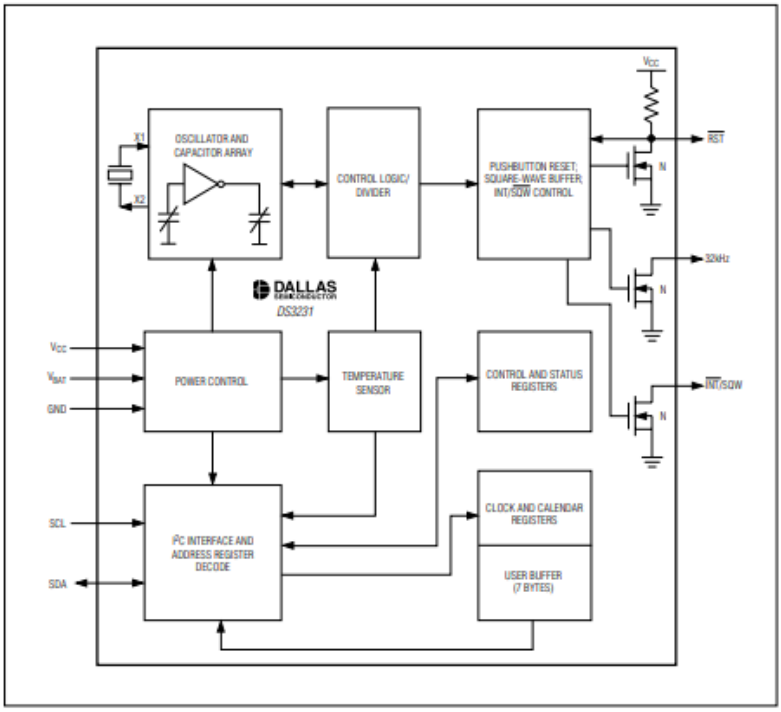

**SUPPLY CURRENT vs. TEMPERATURE**




**FREQUENCY DEVIATION vs. TEMPERATURE vs. AGING VALUE**



## Block Diagram



## Pin Description

| PIN | NAME | FUNCTION |
|---|---|---|
| 1 | 32kHz | 32kHz Output. This open-drain pin requires an external pullup resistor. It may be left open if not used. |
| 2 | $V_{CC}$ | DC Power Pin for Primary Power Supply. This pin should be decoupled using a 0.1µF to 1.0µF capacitor. If not used, connect to ground. |
| 3 | $\overline{INT}$/SQW | Active-Low Interrupt or Square-Wave Output. This open-drain pin requires an external pullup resistor. It may be left open if not used. This multifunction pin is determined by the state of the INTCN bit in the Control Register (0Eh). When INTCN is set to logic 0, this pin outputs a square wave and its frequency is determined by RS2 and RS1 bits. When INTCN is set to logic 1, then a match between the timekeeping registers and either of the alarm registers activates the $\overline{INT}$/SQW pin (if the alarm is enabled). Because the INTCN bit is set to logic 1 when power is first applied, the pin defaults to an interrupt output with alarms disabled. |
| 4 | $\overline{RST}$ | Active-Low Reset. This pin is an open-drain input/output. It indicates the status of $V_{CC}$ relative to the $V_{PF}$ specification. As $V_{CC}$ falls below $V_{PF}$, the $\overline{RST}$ pin is driven low. When $V_{CC}$ exceeds $V_{PF}$, for $t_{RST}$, the $\overline{RST}$ pin is driven high impedance. The active-low, open-drain output is combined with a debounced pushbutton input function. This pin can be activated by a pushbutton reset request. It has an internal 50kΩ nominal value pullup resistor to $V_{CC}$. No external pullup resistors should be connected. If the crystal oscillator is disabled, the startup time of the oscillator is added to the $t_{RST}$ delay. |
| 5–12 | N.C. | No Connection. Must be connected to ground. |
| 13 | GND | Ground |
| 14 | $V_{BAT}$ | Backup Power-Supply Input. This pin should be decoupled using a 0.1µF to 1.0µF low-leakage capacitor. If the I²C interface is inactive whenever the device is powered by the $V_{BAT}$ input, the decoupling capacitor is not required. If $V_{BAT}$ is not used, connect to ground. UL recognized to ensure against reverse charging when used with a lithium battery. Go to www.maxim-ic.com/qa/info/ul. |
| 15 | SDA | Serial Data Input/Output. This pin is the data input/output for the I²C serial interface. This open-drain pin requires an external pullup resistor. |
| 16 | SCL | Serial Clock Input. This pin is the clock input for the I²C serial interface and is used to synchronize data movement on the serial interface. |

## Detailed Description

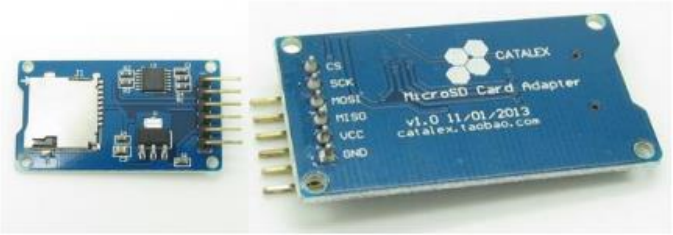
The DS3231 is a serial RTC driven by a temperature-compensated 32kHz crystal oscillator. The TCXO provides a stable and accurate reference clock, and maintains the RTC to within ±2 minutes per year accuracy from -40°C to +85°C. The TCXO frequency output is available at the 32kHz pin. The RTC is a low-power clock/calendar with two programmable time-of-day alarms and a programmable square-wave output. The $\overline{INT}$/SQW provides either an interrupt signal due to alarm conditions or a square-wave output. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The date at the end of the month is automatically adjusted for months with fewer 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. The internal registers are accessible though an I²C bus interface.

A temperature-compensated voltage reference and comparator circuit monitors the level of $V_{CC}$ to detect power failures and to automatically switch to the back-up supply when necessary. The $\overline{RST}$ pin provides an external pushbutton function and acts as an indicator of a power-fail event.

# LAMPIRAN I

## Datasheet Modul Micro SD Adapter

## Micro SD Card Micro SDHC Mini TF Card Adapter Reader Module for Arduino



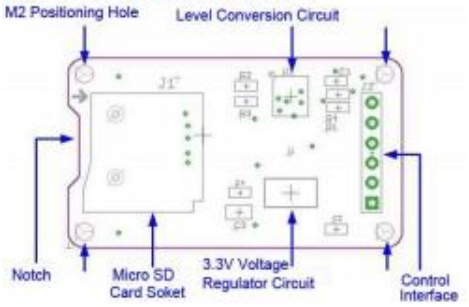

### Description

- The module (MicroSD Card Adapter) is a Micro SD card reader module for reading and writing through the file system and the SPI interface driver, SCM system can be completed within a file MicroSD card
- Support Micro SD Card, Micro SDHC card (high speed card)
- Level conversion circuit board that can interface level is 5V or 3.3V
- Power supply is 4.5V ~ 5.5V, 3.3V voltage regulator circuit board
- Communications interface is a standard SPI interface
- 4 M2 screws positioning holes for easy installation
- Control Interface: A total of six pins (GND, VCC, MISO, MOSI, SCK, CS), GND to ground, VCC is the power supply, MISO, MOSI, SCK for SPI bus, CS is the chip select signal pin;
- 3.3V regulator circuit: LDO regulator output 3.3V for level conversion chip, Micro SD card supply;
- Level conversion circuit: Micro SD card to signal the direction of converts 3.3V, MicroSD card interface to control the direction of the MISO signal is also converted to 3.3V, general AVR microcontroller systems can read the signal;
- Micro SD card connector: self bomb deck, easy card insertion.
- Positioning holes: 4 M2 screws positioning holes with a diameter of 2.2mm, so the module is easy to install positioning, to achieve inter-module combination.

**Interface Parameters:**

| Items | Min | Typical | Max | Unit |
|---|---|---|---|---|
| Power Voltage VCC | 4.5 | 5 | 5.5 | V |
| Current | 0.2 | 80 | 200 | mA |
| Interface Electrical Potential | 3.3 or 5 | | | V |
| Support Card Type | Micro SD Card(<=2G), Mirco SDHC Card(<=32G) | | | — |
| Size | 42X24X12 | | | mm |
| Weight | 5 | | | g |

**Mirco SD Card Interface Module:**



# LAMPIRAN J
## Datasheet Arduino Uno

### Arduino Uno

Arduino Uno R3 Front

Arduino Uno R3 Back

Arduino Uno R2 Front

Arduino Uno SMD

Arduino Uno Front

Arduino Uno Back

### Overview

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.
The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.
Revision 2 of the Uno board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode.
Revision 3 of the board has the following new features:

- 1.0 pinout: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible both with the board that use the AVR, which operate with 5V and with the Arduino Due that operate with 3.3V. The second one is a not connected pin, that is reserved for future purposes.
- Stronger RESET circuit.
- Atmega 16U2 replace the 8U2.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the index of Arduino boards.

### Summary

| | |
|---|---|
| Microcontroller | ATmega328 |
| Operating Voltage | 5V |
| Input Voltage (recommended) | 7-12V |

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

## Schematic & Reference Design

EAGLE files: arduino-uno-Rev3-reference-design.zip (NOTE: works with Eagle 6.0 and newer)
Schematic: arduino-uno-Rev3-schematic.pdf
**Note:** The Arduino reference design can use an Atmega8, 168, or 328, Current models use an ATmega328, but an Atmega8 is shown in the schematic for reference. The pin configuration is identical on all three processors.

## Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.
External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.
The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.
The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.**This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

## Memory

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

## Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.
- **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the analogWrite() function.

- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication using the SPI library.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function. Additionally, some pins have specialized functionality:

- **TWI: A4 or SDA pin and A5 or SCL pin.** Support TWI communication using the Wire library.

There are a couple of other pins on the board:

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

See also the mapping between Arduino pins and ATmega328 ports. The mapping for the Atmega8, 168, and 328 is identical.

## Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).
A SoftwareSerial library allows for serial communication on any of the Uno's digital pins.
The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation for details. For SPI communication, use the SPI library.

## Programming

The Arduino Uno can be programmed with the Arduino software (download). Select "Arduino Uno from the **Tools > Board** menu (according to the microcontroller on your board). For details, see the reference and tutorials.
The ATmega328 on the Arduino Uno comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).
You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details.
The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available . The ATmega16U2/8U2 is loaded with a DFU bootloader, which can be activated by:

- On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2.
- On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode.

You can then use Atmel's FLIP software (Windows) or the DFU programmer (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). See this user-contributed tutorial for more information.

## Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2/16U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.
This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.
The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see this forum thread for details.

### USB Overcurrent Protection

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

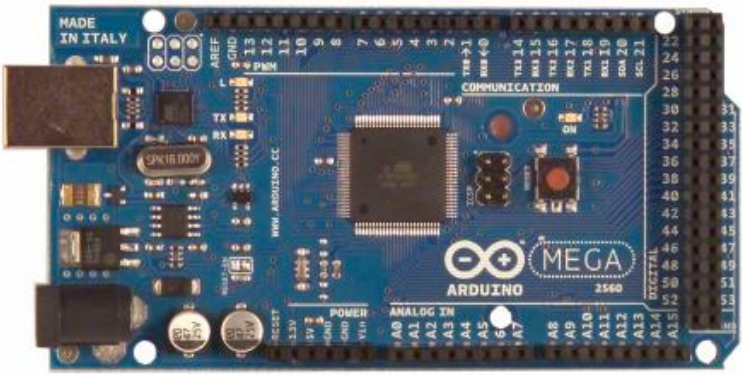
### Physical Characteristics

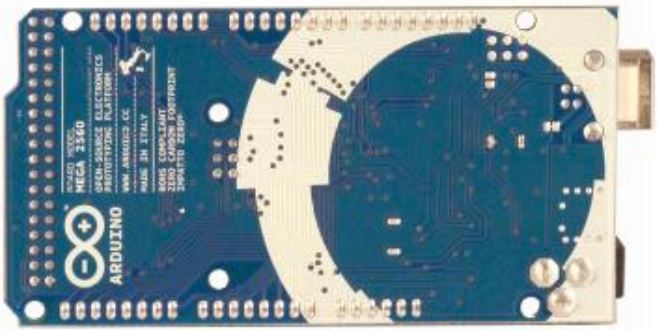
The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Four screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

# LAMPIRAN K
## Datasheet Arduino Mega

ARDUINO_

# Arduino Mega 2560 Datasheet






## Overview

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

## Schematic & Reference Design

EAGLE files: arduino-mega2560-reference-design.zip

Schematic: arduino-mega2560-schematic.pdf

## Summary

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

## Power

The Arduino Mega can be powered via the USB connection or with an external power supply. The power source is selected automatically.

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

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

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

value, the LED is on, when the pin is LOW, it's off.
- **I2C: 20 (SDA) and 21 (SCL).** Support I2C (TWI) communication using the Wire library (documentation on the Wiring website). Note that these pins are not in the same location as the I2C pins on the Duemilanove or Diecimila.

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

There are a couple of other pins on the board:

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

## Communication

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 chip and USB connection to the computer (bu not for serial communication on pins 0 and 1).
A SoftwareSerial library allows for serial communication on any of the Mega2560's digital pins.
The ATmega2560 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation on the Wiring website for details. For SPI communication, use the SPI library.

## Programming

The Arduino Mega can be programmed with the Arduino software (download). For details, see the reference and tutorials.
The ATmega2560 on the Arduino Mega comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It

communicates using the original STK500 protocol (reference, C header files).
You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details.

## Automatic (Software) Reset

Rather then requiring a physical press of the reset button before an upload, the Arduino Mega2560 is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega2560 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.
This setup has other implications. When the Mega2560 is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Mega2560. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.
The Mega2560 contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see this forum thread for details.

## USB Overcurrent Protection

The Arduino Mega2560 has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

# LAMPIRAN L
## Datasheet TFT LCD



### 1. General Description

This display module is a transmissive type color active matrix TFT(Thin Film Transistor) liquid crystal display (LCD) that uses amorphous silicon TFT as a switching device. This module is composed of a TFT LCD module, a driver circuit, and a back-light unit.
The resolution of a 2.4" contains 240（RGB)X320 dots and can display up to 262k colors.

### 2. Module Parameter

| Features | Details | Unit |
|---|---|---|
| Display Size(Diagonal) | 2．4" | - |
| LCD type | a-Si TFT | - |
| Support interface mode | MCU SPI RGB | - |
| Display Mode | TN/ Normally white | - |
| Resolution | 240 RGB x 320 | - |
| View Direction | 12 O'clock | Best image |
| Grayscale Inversion Direction | 6 O'clock | - |
| Module Outline | 42.72(H) ×60.26(V)×2.2 (T) | mm |
| TP Outline | 42.32(H) x59.26(V)x1.0(T) | mm |
| TP Viewing Area | 38.72(H) x47.92(V) | mm |
| TP Active Area | 37.72(H) x48.36(V) | mm |
| Active Area | 36.72 (H)X 48.96(V) | mm |
| Viewing Area | N/A | mm |
| Pixel Size | 0.153(H) x0.153 (V) | mm |
| Pixel Arrangement | Stripe | - |
| Display Colors | 262K | - |
| Interface | System parallel interface | - |
| Driver IC | ILI9341 | - |
| Operating Temperature | -20~70 | °C |
| Storage Temperature | -30~80 | °C |
| LCM brightness | 210 | $cd/m^2$ |
| Weight | TBD | g |

### 3. Absolute Maximum Ratings

$V_{SS}$=0V, Ta=25°C

| Item | | Symbol | Min. | Max. | Unit |
|---|---|---|---|---|---|
| Supply Voltage | Power supply | VCC | -0.3 | +4.6 | V |
| | Analog | VCI | - | - | V |
| | IO | IOVCC | - | - | V |
| Input Voltage | | VI | -0.3 | IOVCC+0.3 | V |
| Storage temperature | | $T_{stg}$ | -30 | +80 | °C |
| Operating temperature | | $T_{op}$ | -20 | +70 | °C |
| Storage humidity | | $H_{stg}$ | 10 | Note 1 | %RH |
| Operating humidity | | $H_{op}$ | 10 | Note 1 | %RH |

Note 1: 90%RH max. If Ta is below 50°C; 60%RH max. If Ta is over 60°C.

## 4. DC Characteristics

| Item | | Symbol | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| Supply Voltage | Power supply | VCC | 2.5 | 2.8 | 3.3 | V |
| | Analog | VCI | 2.5 | 2.8 | 3.3 | V |
| | IO | IOVCC | 1.65 | 2.8 | 3.3 | V |
| Logic Low input voltage | | $V_{IL}$ | 0.0 | - | 0.2*IOVCC | V |
| Logic High input voltage | | $V_{IH}$ | 0.7*IOVCC | - | IOVCC | V |
| Logic Low output voltage | | $V_{OL}$ | - | - | 0.2*IOVCC | V |
| Logic High output voltage | | $V_{OH}$ | 0.8*IOVCC | - | IOVCC | V |
| Current Consumption | Normal display | Ivdd | - | - | - | mA |
| | Standby mode | Ivdd- | - | - | - | uA |
| Frame Frequency | | $f_{FR}$ | - | TBD | - | Hz |

## 5. Backlight Characteristics

### 5.1. Backlight Characteristics

| Item | Symbol | Condition | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| Forward Voltage | $V_f$ | Ta=25 °C, $I_F$=(15*4)mA | 3.1 | 3.2 | 3.3 | V/LED |
| Forward Current | $I_f$ | Ta=25 °C, $V_F$=3.2V | - | 60 | - | mA |
| Luminance | $L_v$ | - | | 3500 | | $cd/m^2$ |
| Uniformity | Avg | - | - | - | - | % |
| CIE | X | - | | - | - | - |
| | Y | - | - | - | - | - |
| Power dissipation | $P_d$ | - | - | - | - | $mW$ |
| Backlight Driving Voltage | $V_{AK}$ | - | - | - | - | V |
| Drive method | Constant current | | | | | |
| LED Configuration | 4 White LEDs in Parallel | | | | | |

**Note:** Test condition $I_f$ =60mA, Ta=25°C.

## 6. Optical Characteristics

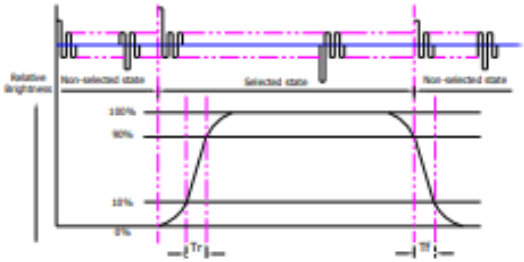
### 6.1. Optical Characteristics

Ta=25°C, VDD=2.8V, TN LC+ Polarizer

| | Item | | Symbol | Condition | Specification | | | Unit |
|---|---|---|---|---|---|---|---|---|
| | | | | | Min. | Typ. | Max. | |
| Backlight On (Transmissive Mode) | Luminance on TP surface($I_f$=60mA) | | Lv | Normally viewing angle $\theta_x=\theta_y$ =0° | - | -TBD | - | cd/m² |
| | Contrast ratio(See 6.3) | | CR | | 400 | 500 | - | - |
| | Response time (See 6.2) | | TR+TF | | - | 16 | 32 | ms |
| | Chromaticity Transmissive (See 6.5) | Red | XR | - | 0.606 | 0.626 | 0.646 | - |
| | | | YR | | 0.314 | 0.334 | 0.354 | - |
| | | Green | XG | | 0.299 | 0.319 | 0.339 | - |
| | | | YG | | 0.537 | 0.557 | 0.577 | - |
| | | Blue | XB | | 0.122 | 0.142 | 0.162 | - |
| | | | YB | | 0.102 | 0.122 | 0.142 | - |
| | | White | XW | | 0.298 | 0.318 | 0.338 | - |
| | | | YW | | 0.317 | 0.337 | 0.357 | - |
| | Viewing Angle (See 6.4) | Horizontal | θx+ | Center CR≥10 | - | 45 | - | Deg. |
| | | | θx- | | - | 45 | - | |
| | | Vertical | θy+ | | - | 45 | - | |
| | | | θy- | | - | 20 | - | |
| | NTSC Ratio(Gamut) | | - | - | - | TBD | - | % |

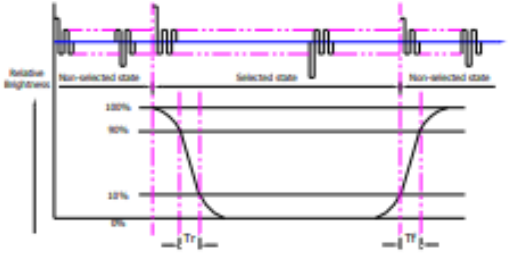
### 6.2. Definition of Response Time

6.2.1. Normally Black Type (Negative)




Tr is the time it takes to change form non-selected state with relative luminance 10% to selected state with relative luminance 90%;
Tf is the time it takes to change from selected state with relative luminance 90% to non-selected state with relative luminance 10%.

6.2.2. Normally White Type (Positive)




Tr is the time it takes to change form non-selected state with relative luminance 90% to selected state with relative luminance 10%;
Tf is the time it takes to change from selected state with relative luminance 10% to non-selected state with relative luminance 90%;

### 6.3. Definition of Contrast Ratio

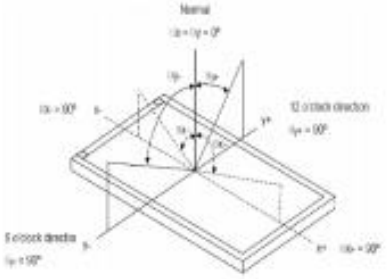
Contrast is measured perpendicular to display surface in reflective and transmissive mode.

The measurement condition is:

| | |
|---|---|
| Measuring Equipment | BM-7 or EQUI |
| Measuring Point Diameter | 3mm/1mm |
| Measuring Point Location | Active Area centre point |
| Test pattern | A: All Pixels white |
| | B: All Pixel black |
| Contrast setting | Maximum |

Definitions: CR (Contrast) = Luminance of White Pixel / Luminance of Black Pixel

### 6.4. Definition of Viewing Angles



Measuring machine: LCD-5100 or EQUI

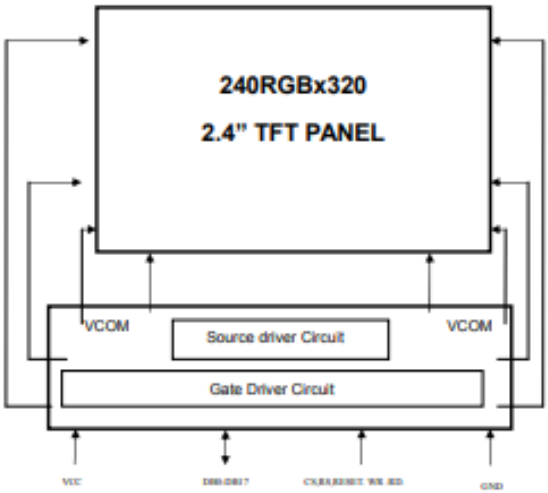
### 6.5. Definition of Color Appearance

R,G,B and W are defined by (x, y) on the IE chromaticity diagram

NTSC=area of RGB triangle/area of NTSC triangleX100%

Measuring picture: Red, Green, Blue and White (Measuring machine: BM-7)

## 7. Block Diagram and Power Supply



## 8. Interface Pins Definition

### 8.1. Module interface

| No. | Symbol | I/O | DESCRIPTION |
|---|---|---|---|
| 1 | VCI | P | Power Supply |
| 2 | IOVCC | P | Power supply for IO port |
| 3 | IM0 | I | Select the MCU interface mode<br>MPU Parallel interface bus and serial interface select<br>If use RGB Interface must select serial interface. (Note1). |
| 4 | IM1 | | |
| 5 | IM2 | | |
| 6 | IM3 | | |
| 7 | RESET | I | LCM reset signal |
| 8 | VSYNC | I | Frame synchronizing signal for RGB interface operation. |
| 9 | HSYNV | I | Line synchronizing signal for RGB interface operation. |
| 10 | DOTCLK | I | Dot clock signal for RGB interface operation. |
| 11 | ENABLE | I | Data enable signal for RGB interface operation. |
| 12^29 | DB17^DB0 | I/P | Data bus |
| 30 | SDO | I/P | LCD Read for the MPU interface |
| 31 | SDI | I | Write control pin for the MPU interface |
| 32 | RD | I | LCD Read for the MPU interface |
| 33 | WR | I | Write control pin for the MPU interface |
| 34 | RS/SCL | I | ( Please according to, the IC specification choice model). |
| 35 | CS | I | The data is applied on the rising edge of the SCL signal.If not used, fix this pin at VDDI or VSS. |
| 36 | GND | P | Ground |
| 37 | LEDA | I | LED Anode |
| 38 | LEDK1 | I | LED Cathode |
| 39 | LEDK2 | | |
| 40 | LEDK3 | | |
| 41 | LEDK4 | | |
| 42 | XR | I | TOUCH P ANLE PIN |
| 43 | YD | | |
| 44 | XL | | |
| 45 | YU | | |
| | | | |

# LAMPIRAN M
## Dokumentasi Tugas Akhir

Pencucian Ikan Teri dengan Air Garam

Penirisan Ikan Teri

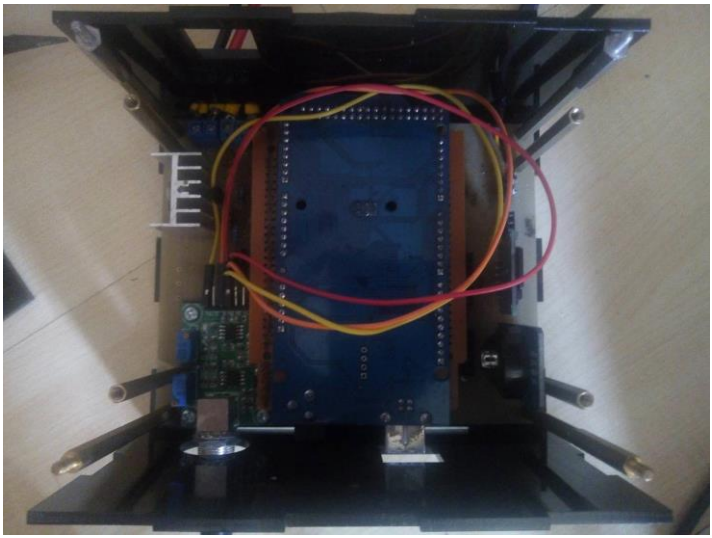
Pemberian Garam pada Ikan Teri

Penambahan Gula Merah pada Ikan Teri

Pengadukan Ikan, Gula dan Garam

Penirisan Ikan Teri Benar-Benar Kering dengan Tissue

Realisasi Sistem Interface



Realisasi Sistem Kontrol

Realisasi Bagian Dalam Pemanas Kawat Nikrom

Realisasi Pemanas Kawat Nikrom dengan Penambahan Sterofoam sebagai Isolator Pemusatan Perambatan Panas

Realisasi Box Perangkat Elektronik

Tampak Atas Realisasi Keseluruhan Alat

Tampak Depan Realisasi Keseluruhan Alat

Tampak Kiri Realisasi Keseluruhan Alat

Tampak Belakang Realisasi Keseluruhan Alat

Tampak Kanan Realisasi Keseluruhan Alat

Data Fermentasi Hari ke 0

Data Fermentasi Hari ke 2

Data Fermentasi Hari ke 5

Fermentasi Hari ke 8

Fermentasi Hari ke 10

Akhir Fermentasi Hari ke 10

Hasil Akhir Fermentasi Rusip

Pengemasan Rusip untuk Pengiriman ke Responden Human Sensory Test

Bentuk Gambaran Rusip yang Dijual secara Komersil

## BIODATA PENULIS

Penulis buku ini memiliki nama lengkap Axel William, dengan nama panggilan Axel lahir di Jakarta pada tanggal 18 Mei 1998 merupakan anak pertama dari dua bersaudara. Penulis menyelesaikan pendidikan dasar di SDK Santa Angelina (2004-2010), dilanjutkan pendidikan tingkat menengah di SMPK Sang Timur (2010-2013), dan sekolah tingkat atas di SMAK Penabur Bintaro Jaya (2013-2016). Setelah Menyelesaikan Studi menengah atas penulis memutuskan untuk merantau dari kota asal Tangerang menuju Surabaya untuk menempuh gelar sarjana di Departemen Teknik Elektro, Fakultas Teknologi Elektro dan Informatika Cerdas, Institut Teknologi Sepuluh Nopember Surabaya. Penulis tergolong sebagai seseorang yang visioner, seseorang yang menjalankan segala sesuatu sesuai dengan rencana yang sudah dirancang secara matang dari jauh hari, dan dalam jangka waktu dekat penulis memiliki keinginan kuat bagi untuk melanjutkan studi pasca sarjana di luar negeri.


| | |
|---|---|
| Email | : axel.william@gmail.com |
| Hp/WA | : 087808412953 |
| Facebook | : Axel William |