



TUGAS AKHIR - EE 184801

**KONTROL KELEMBAPAN TANAH MENGGUNAKAN
KAMERA TERMAL DAN *FUZZY LOGIC***

Ricky Julio Rajagukguk
NRP 0711154000076

Dosen Pembimbing
Dr. Muhammad Rivai, ST. MT.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2019



TUGAS AKHIR - EE 184801

**KONTROL KELEMBAPAN TANAH MENGGUNAKAN
KAMERA TERMAL DAN *FUZZY LOGIC***

Ricky Julio Rajagukguk
NRP 0711154000076

Dosen Pembimbing
Dr. Muhammad Rivai, ST.MT.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2019



FINAL PROJECT - EE 184801

**SOIL MOISTURE CONTROL USING THERMAL CAMERA
AND *FUZZY LOGIC***

Ricky Julio Rajagukguk
NRP 0711154000076

Dosen Pembimbing
Dr. Muhammad Rivai, ST. MT.

ELECTRICAL ENGINEERING DEPARTMENT
Faculty of Electrical Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2019

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul "Kontrol Kelembapan Tanah Menggunakan Kamera Termal dan *Fuzzy Logic*" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan 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, Mei 2019



Ricky Julio Rajagukguk
NRP. 0711 15 4000 0076

.....*Halaman ini sengaja dikosongkan*.....

KONTROL KELEMBAPAN TANAH MENGUNAKAN KAMERA TERMAL DAN *FUZZY* LOGIC

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik
Pada
Bidang Studi Elektronika
Departemen Teknik Elektro
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember**

**Menyetujui :
Dosen Pembimbing**

**Dr. Muhammad Rival, ST, MT
NIP. 196904261994031003**

SURABAYA

JULI, 2019

.....*Halaman ini sengaja dikosongkan*.....

KONTROL KELEMBAPAN TANAH MENGUNAKAN KAMERA TERMAL DAN *FUZZY LOGIC*

Nama : Ricky Julio Rajagukguk
Pembimbing : Dr. Muhammad Rivai, S.T.,M.T.

ABSTRAK

Kelembapan tanah merupakan faktor penting dalam industri pertanian dan perkebunan. Tanah dengan kelembapan yang cukup akan sangat mempengaruhi produksi dari tanaman itu sendiri. Kurangnya ketersediaan air menjadi masalah serius yang dihadapi, dimana saat musim kemarau hanya mengandalkan curah hujan karena kurangnya pasokan air. Hal ini dapat diatasi dengan melakukan pengontrolan terhadap jumlah air yang dikeluarkan. Penyiraman tanaman dengan metode lama kurang efektif, karena hanya mengandalkan perspektif dari petani, tidak berdasarkan keadaan tanah yang sebenarnya. Pada tugas akhir ini, akan dibuat suatu sistem kontrol kelembapan tanah menggunakan kamera termal dan fuzzy logic. Sensor yang digunakan merupakan sensor inframerah atau termal yang beroperasi dengan tegangan input 3-5V, dengan array 8x8 (64 piksel). Ketika terhubung ke mikrokontroler arduino, ia akan menampilkan array dari 64 pembacaan suhu inframerah. Sensor ini akan mengukur suhu mulai dari 0⁰ C hingga 80⁰ C (32⁰ F hingga 176⁰ F) dengan akurasi ±2.5⁰ C(4.5⁰ F), dapat mendeteksi manusia hingga jarak 7 meter (23) kaki, dengan frame rate maksimum 10Hz. Dengan menggunakan sensor ini, kelembapan tanah dapat diketahui tanpa ada kontak langsung dari alat dengan tanah, sehingga lebih praktis dan tidak mudah karatan. Kontrol kelembapan ini akan mengontrol seberapa lama air dialirkan ke tanah, sampai sensor mendeteksi bahwa kelembapan tanah sudah cukup, maka penyiraman tanah akan diberhentikan.

Kata Kunci : Fuzzy Logic, Kamera Termal, Tanah

.....*Halaman ini sengaja dikosongkan*.....

SOIL MOISTURE CONTROL USING THERMAL CAMERA AND FUZZY LOGIC

Name : Ricky Julio Rajagukguk
Supervisor : Dr. Muhammad Rivai, S.T.,M.T.

ABSTRACT

Soil moisture is an important factor in the agriculture and plantation industries. Soil with sufficient moisture will greatly affect the production of the plant itself. Lack of water availability is a serious problem faced, where during the dry season only rely on rainfall due to lack of water supply. This can be overcome by controlling the amount of water released. Watering the plants with the old method is less effective, because it only relies on the perspective of the farmers, not based on the actual soil conditions. In this final project, a soil moisture control system will be created using thermal and fuzzy logic cameras. The sensors used are infrared or thermal sensors that operate with 3-5V input voltages, with 8x8 arrays (64 pixels). When connected to an Arduino microcontroller, it displays an array of 64 infrared temperature readings. This sensor will measure temperatures ranging from 00 C to 800 C (320 F to 1760 F) with an accuracy of ± 2.50 C (4.50 F), can detect humans up to a distance of 7 meters (23) feet, with a maximum frame rate of 10Hz. By using this sensor, soil moisture can be known without direct contact from the tool with the ground, making it more practical and not rusty. This humidity control will control how long the water is flowed to the ground, until the sensor detects that the soil moisture is sufficient, then the watering of the soil will be stopped.

Keywords: Fuzzy Logic, Thermal Camera, Soil

.....*Halaman ini sengaja dikosongkan*.....

KATA PENGANTAR

Puji dan syukur kepada Tuhan Yesus atas berkat dan penyertaannya yang, penulis dapat menyelesaikan laporan tugas akhir dengan judul **“KONTROL KELEMBAPAN TANAH MENGGUNAKAN KAMERA TERMAL DAN FUZZY LOGIC”**, sebagai salah satu persyaratan dalam menyelesaikan pendidikan program Strata-Satu di Departemen Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember.

Dalam penyusunan dan penyelesaian laporan Tugas Akhir ini penulis mendapatkan banyak sekali doa, bantuan dan dukungan dari berbagai pihak. Untuk itu penulis mengucapkan terima kasih yang sebesar – besarnya kepada :

1. Bapak Dr. Muhammad Rivai, S.T., M.T. selaku dosen pembimbing atas gagasan topik tugas akhir serta bimbingan dan arahan untuk penulis selama mengerjakan tugas akhir.
2. Bapak Astria Nur Irfansyah, ST., M.Eng., Dr. , Ir. Harris Pirngadi, M.T., Fajar Budiman, ST., M.Sc., Ir. Tasripan, M.T. selaku dosen penguji pada ujian Tugas Akhir saya
3. Bapak Dr. Eng. Ardyono Priyadi, S.T.,M.Eng. selaku Kepala Departemen Teknik Elektro ITS.
4. Kepada seluruh dosen dan staf pegawai Departemen Teknik Elektro ITS.
5. Kedua orang tua, Bapak dan Mama yang sudah membesarkan penulis, dan terus membantu perkuliahan dari awal masuk sampai hari ini.
6. Helena Maretha Rajagukguk dan Naomi Januari Rajagukguk, yang telah menambah motivasi kuliah, dan tidak jarang memberikan suntikan dana segar.
7. Oyen, yang secara tiba-tiba hadir dan merubah segalanya.
8. Tim Semeru, Markicabs Sempak Basah, dan Pendaki Cantik yang telah menemani mendaki gunung selama suntuk perkuliahan, dan memberikan banyak pembelajaran menarik buat penulis.
9. Penghuni Rumah Yudas sebagai teman seataap, sekamar mandi, dan banyak lagi seper-seperan, yang kalau disebutkan semua, nantinya buku ini tidak akan cukup.
10. Nisita Hapsari yang sudah menjadi kakak penulis selama di

Surabaya, memberikan cukup banyak pelajaran buat penulis.

11. Seluruh teman-teman NHKBP Manyar yang menjadi teman sepelayanan, sepercopasan, dan juga menjadi tempat pembelajaran yang sangat banyak untuk penulis.
12. Tim Hore yang menjadi tempat untuk TOD, dimana hal ini sangat menyenangkan buat penulis.
13. Penghuni Lab Robot Cerdas yang saya yakin penghuninya tidak kalah cerdas dengan robotnya.
14. Teman-teman bidang studi elektronika yang tidak dapat disebutkan satu-persatu yang telah membantu dan memberikan semangat kepada penulis selama menjalani perkuliahan di Departemen Teknik Elektro ITS.

Penulis menyadari bahwa masih banyak yang dapat dikembangkan pada tugas akhir ini. Oleh karena itu penulis menerima setiap masukan dan kritik yang diberikan. Semoga tugas akhir ini dapat memberikan manfaat bagi banyak pihak.

Surabaya, Mei 2019

Ricky Julio Rajagukguk

DAFTAR ISI

PERNYATAAN KEASLIAN	i
LEMBAR PENGESAHAN	iii
ABSTRAK	v
<i>ABSTRACT</i>	vii
KATA PENGANTAR.....	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xviii
PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Perumusan Masalah	2
1.3. Tujuan Penelitian	2
1.4. Batasan Masalah	2
1.5. Metodologi Penelitian	2
1.6. Sistematika Penulisan	4
1.7. Relevansi	4
TINJAUAN PUSTAKA	7
2.1. Tanah.....	7
2.1.1. Tekstur Tanah	7
2.1.2. Struktur Tanah	9
2.1.3. Klasifikasi Kelembapan Tanah.....	9
2.2. Radiasi Benda Hitam dan Spektrum Inframerah.....	10
2.3. Kamera Termal	12
2.4. Infrared Array Sensor Grid-Eye (AMG8833).....	13
2.5. Arduino	14
2.5.1. Pin Power	15
2.5.2. Input dan Output.....	16
2.5.3. Pin Komunikasi	16
2.5.4. Pemrograman	17
2.6. Fuzzy Logic.....	17
2.6.1. Arsitektur Fuzzy Logic	17
2.6.2. Fungsi Keanggotaan	18
2.7. LCD (Liquid Crystal Display).....	19
2.7.1. Dasar Teori LCD	19
2.7.2. Prinsip Kerja LCD 16x2	20
2.7.2. Deskripsi Pin LCD 16x2.....	20

2.8.	I2C.....	21
2.9.	RTC DS3231.....	22
2.10.	HC-05.....	23
2.11.	Solenoid Valve.....	24
2.12.	Relay.....	26
PERANCANGAN SISTEM.....		29
3.1.	Diagram Blok Sistem.....	29
3.2.	Perancangan Perangkat Keras.....	31
3.2.1.	Desain Tandon.....	31
3.2.2.	LCD dengan I2C.....	31
3.2.3.	Pembacaan sensor AMG8833.....	32
3.2.4.	HC - 05.....	33
3.2.5.	RTC (Real Time Clock).....	34
3.2.6.	Relay dan <i>Solenoid Valve</i>	35
3.2.7.	Arduino Uno.....	36
3.3.	Perancangan Perangkat Lunak.....	36
3.3.1.	Pembacaan Sensor AMG8833.....	37
3.3.2.	Sistem Kontrol Fuzzy Logic.....	38
PENGUJIAN DAN ANALISIS.....		43
4.1.	Kalibrasi AMG8833 dengan Sensor Kelembapan.....	43
4.1.1.	Kalibrasi di Dalam Ruangan.....	44
4.1.2.	Kalibrasi di Lahan.....	47
4.2.	Pengujian Sensor AMG8833.....	49
4.2.1.	Pengujian di Dalam Ruangan.....	49
4.2.2.	Pengujian di Lahan.....	54
4.3.	Pengujian Sistem Fuzzy Logic.....	68
4.3.1.	Pengujian di Dalam Ruangan.....	68
4.3.2.	Pengujian di Lahan.....	71
4.4.	Pengujian Keseluruhan Sistem.....	74
PENUTUP.....		83
5.1.	Kesimpulan.....	83
5.2.	Saran.....	83
DAFTAR PUSTAKA.....		85
LAMPIRAN A.....		89
LAMPIRAN B.....		107
LAMPIRAN C.....		111
LAMPIRAN D.....		115
LAMPIRAN E.....		123
BIODATA PENULIS.....		137

DAFTAR GAMBAR

Gambar 2.1 Tanah Produktif	7
Gambar 2.2 Tanah Liat	8
Gambar 2.3 Tanah Pasir	8
Gambar 2.4 Klasifikasi Kelembapan Tanah	10
Gambar 2.5 Kurva Radiasi Benda Hitam	11
Gambar 2.6 Spektrum Gelombang Inframerah	11
Gambar 2.7 Kamera Termal	12
Gambar 2.8 Sensor AMG8833	13
Gambar 2.9 Rangkaian Internal Sensor AMG8833	13
Gambar 2.10 Arduino Uno	14
Gambar 2.11 Arduino Uno	15
Gambar 2.12 Arsitektur Fuzzy	18
Gambar 2.13 Contoh Fungsi Keanggotaan	19
Gambar 2.14 LCD 16x2	20
Gambar 2.15 I2C	21
Gambar 2.16 Pin Diagram dari DS3231	22
Gambar 2.17 Modul RTC DS3231	23
Gambar 2.18 Modul Bluetooth HC-05	24
Gambar 2.19 Solenoid Valve	25
Gambar 2.20 Cara Kerja Solenoid Valve	25
Gambar 2.21 Skematik Solenoid Valve	25
Gambar 2.22 Bentuk dan Simbol Relay	26
Gambar 2.23 Struktur Sederhana Relay	26
Gambar 3.1 Skema Sistem Keseluruhan	29
Gambar 3.2 Diagram Blok Sistem	30
Gambar 3.3 Desain Tandon	31
Gambar 3.4 LCD 16x2 terhadap I2C	32
Gambar 3.5 Antarmuka LCD 16x2 I2C dengan Arduino Uno	32
Gambar 3.6 Antarmuka Sensor AMG8833 dengan Arduino Uno	33
Gambar 3.7 Antarmuka HC-05 dengan Arduino Uno	34
Gambar 3.8 Antarmuka RTC dengan Arduino Uno	35
Gambar 3.9 Antarmuka Relay dan Solenoid Valve dengan Arduino Uno	36
Gambar 3.10 Flowchart Sistem Keseluruhan	37
Gambar 3.11 Flowchart Sistem Fuzzy Logic	39
Gambar 3.12 Fungsi Keanggotaan error di Dalam Ruang	39
Gambar 3.13 Fungsi Keanggotaan Δ error di Dalam Ruang	40

Gambar 3.14 Fungsi Keanggotaan error Saat Siang Hari	40
Gambar 3.15 Fungsi Keanggotaan Δ error Saat Siang Hari	41
Gambar 3.16 Fungsi Keanggotaan Output	41
Gambar 4.1 Gambar Realisasi Alat di Lahan.....	43
Gambar 4.2 Sensor Kelembapan Tanah Analog	44
Gambar 4.3 Kalibrasi Tanah di Dalam Ruangan dengan Kelembapan yang Berbeda	45
Gambar 4.4 Hubungan Antara Kelembapan dengan Kamera Termal	46
Gambar 4.5 Kalibrasi Tanah di Lahan dengan Kelembapan yang Berbeda	47
Gambar 4.6 Hubungan Antara Kelembapan dengan Kamera Termal	48
Gambar 4.7 Sampel Tanah dengan Kelembapan yang Berbeda	49
Gambar 4.8 Hasil Pembacaan Sensor di Dalam Ruangan	50
Gambar 4.9 Grafik Rata-rata Suhu untuk Tanah Kering	51
Gambar 4.10 Grafik Rata-rata Suhu untuk Tanah Agak Basah	51
Gambar 4.11 Grafik Rata-rata Suhu untuk Tanah Basah	51
Gambar 4.12 Hasil Pembacaan Sensor untuk Tanah Basah dengan Air Biasa.....	52
Gambar 4.13 Hasil Pembacaan Sensor untuk Tanah Basah dengan Air Panas	52
Gambar 4.14 Hasil Pembacaan Sensor untuk Tanah Basah dengan Air Dingin.....	53
Gambar 4.15 Posisi dan Jarak pada Pengujian AMG8833	54
Gambar 4.16 Keadaan Tanah Saat Kering	55
Gambar 4.17 Hasil Pembacaan Sensor Saat Tanah Kering	56
Gambar 4.18 Keadaan Tanah Saat Agak Basah	56
Gambar 4.19 Hasil Pembacaan Sensor Saat Tanah Agak Basah	57
Gambar 4.20 Keadaan Tanah Saat Basah	57
Gambar 4.21 Hasil Pembacaan Sensor Saat Tanah Basah.....	58
Gambar 4.22 Keadaan Tanah Saat Kering	59
Gambar 4.23 Hasil Pembacaan Sensor Saat Tanah Kering	59
Gambar 4.24 Keadaan Tanah Saat Agak Basah	60
Gambar 4.25 Hasil Pembacaan Sensor Saat Tanah Agak Basah	60
Gambar 4.26 Keadaan Tanah Saat Basah	61
Gambar 4.27 Hasil Pembacaan Sensor Saat Tanah Basah.....	62
Gambar 4.28 Keadaan Tanah Saat Kering	63
Gambar 4.29 Hasil Pembacaan Sensor Saat Tanah Kering	63
Gambar 4.30 Keadaan Tanah Saat Agak Basah	64
Gambar 4.31 Hasil Pembacaan Sensor Saat Tanah Agak Basah	64

Gambar 4.32 Keadaan Tanah Saat Basah.....	65
Gambar 4.33 Hasil Pembacaan Sensor Saat Tanah Basah	66
Gambar 4.34 Grafik Hasil Pembacaan Sensor dari Kondisi Kering ke Kondisi Basah di Pagi Hari.....	66
Gambar 4.35 Grafik Hasil Pembacaan Sensor dari Kondisi Kering ke Kondisi Basah di Siang Hari.....	67
Gambar 4.36 Grafik Hasil Pembacaan Sensor dari Kondisi Kering ke Kondisi Basah di Sore Hari	67
Gambar 4.37 Pengujian di Dalam Ruangan	68
Gambar 4.38 Hasil Pembacaan Sensor Saat Tanah Kering	69
Gambar 4.39 Hasil Pembacaan Sensor Saat Tanah Agak Basah	70
Gambar 4.40 Hasil Pembacaan Sensor Saat Tanah Basah	70
Gambar 4.41 Valve Terbuka	71
Gambar 4.42 Hasil Pembacaan Sensor Sebelum Disiram	72
Gambar 4.43 Hasil Pembacaan Sensor Setelah 3 Detik Penyiraman	72
Gambar 4.44 Hasil Pembacaan Sensor Setelah 5 Detik Penyiraman	73
Gambar 4.45 Hasil Pembacaan Sensor Setelah 8 Detik Penyiraman	73
Gambar 4.46 Keseluruhan Sistem	74
Gambar 4.47 Hasil Pada LCD	74
Gambar 4.48 Bagian Receiver.....	75
Gambar 4.49 Hasil Pembacaan Sensor Saat Tanah Kering.....	75
Gambar 4.50 Hasil Pembacaan Sensor Setelah Dilakukan Penyiraman	76
Gambar 4.51 Hasil Pembacaan Sensor di Dalam Ruangan	78
Gambar 4.52 Hasil Pembacaan Sensor di Lahan.....	79
Gambar 4.53 Alat Menyiram Tanah yang Kering.....	80
Gambar 4.54 Hasil Pembacaan Sensor Analog	81

.....*Halaman ini sengaja dikosongkan*.....

DAFTAR TABEL

Tabel 2.1. Spesifikasi Arduino Uno	14
Tabel 2.2. Deskripsi Pin pada LCD 16x2	20
Tabel 2.1. Deskripsi Pin pada RTC DS3231	23
Tabel 3.1. Spesifikasi Sensor AMG8833	33
Tabel 3.2. Spesifikasi HC-05	34
Tabel 3.3. Spesifikasi RTC	34
Tabel 3.4. Aturan Dasar (Rule Base)	42
Tabel 4.1. Percobaan dengan Kelembapan yang Berbeda	46
Tabel 4.2. Percobaan dengan Kelembapan yang Berbeda	48
Tabel 4.3. Hasil Pengujian Sistem Fuzzy Logic di Dalam Ruangan	69
Tabel 4.4. Hasil Pengujian Sistem Fuzzy Logic pada Siang Hari	71
Tabel 4.5. Pengujian Sensor dalam Mendeteksi Kelembapan	76
Tabel 4.6. Pengujian Sensor dalam Mendeteksi Kelembapan di Lahan	78
Tabel 4.5. Pengujian Pengontrolan Kelembapan Tanah	80

.....*Halaman ini sengaja dikosongkan*.....

BAB I

PENDAHULUAN

Tugas akhir merupakan suatu penelitian yang dilakukan sebagai persyaratan akademik untuk mendapatkan gelar sarjana teknik di Institut Teknologi Sepuluh Nopember (ITS) Surabaya. Topik yang akan dibahas pada tugas akhir ini mengenai pemanfaatan *Internet of Things* pada alat *Vending Machine*.

Pada bab ini membahas mengenai hal-hal yang mendahului pelaksanaan tugas akhir. Hal tersebut meliputi latar belakang, perumusan masalah, tujuan penulisan, batasan masalah, metodologi penelitian, sistematika penulisan, dan relevansi.

1.1. Latar Belakang

Tanah adalah campuran mineral dan bahan organik yang mengandung udara, air, dan mikroorganisme. Tanah merupakan media di mana tanaman tumbuh, habitat untuk hewan, dan sebagai tempat penyimpanan air. Banyak proses yang terjadi di tanah, termasuk daur ulang nutrisi, pemurnian air, dan pertukaran gas dengan atmosfer. Kandungan mineral tanah berasal dari bahan induk, umumnya batuan dasar, yang telah dimodifikasi dari waktu ke waktu melalui proses fisik, kimia dan biologi [1].

Kelembapan tanah adalah air yang mengisi sebagian atau seluruh pori – pori tanah yang berada di atas *water table*. Definisi yang lain menyebutkan bahwa kelembapan tanah menyatakan jumlah air yang tersimpan di antara pori – pori tanah, dimana kelembapan tanah sangat dinamis, hal ini disebabkan oleh penguapan melalui permukaan tanah, transpirasi dan perkolasi [2]. Jika kita dapat mengetahui tingkat kelembapan tanah pada suatu tempat tertentu, maka data tersebut dapat digunakan untuk beberapa aplikasi, yaitu memantau kekeringan, membantu produktivitas tanaman, memprediksi banjir, perkiraan cuaca, dan menghubungkan siklus air, energi, dan karbon [3].

Pada definisi tanah sebelumnya, kita catat bahwa tanah digunakan sebagai medium tempat tumbuh tanaman. Akar dapat menembus tanah sehingga tanaman dapat menanamkan dirinya dan juga dapat mengakses nutrisi dan air yang tersimpan di dalam tanah.

Ketersediaan air yang disimpan dalam tanah sangat penting untuk pertumbuhan tanaman. Jika kandungan air yang terdapat dalam tanah terlalu sedikit, atau dengan kata lain kelembapan tanah turun di bawah titik layu, tanaman berjuang untuk bertahan hidup. Demikian pula, terlalu banyak air bisa sama fatalnya [4].

Kontrol kelembapan tanah menggunakan kamera termal merupakan solusi dari kurang efektifnya kontrol kelembapan tanah yang dilakukan dengan menggunakan sensor yang ditanam, dimana sensor akan mudah rusak karena bersentuhan langsung dengan tanah, atau secara konvensional, dimana pengontrolan dilakukan berdasarkan intuisi dari pemilik lahan. Sensor kamera termal dapat mendeteksi daerah yang lebih luas, berbeda dengan sensor tanam yang harus ditanam di setiap titik untuk mengetahui kelembapan dari tanah yang diinginkan, kamera termal dapat mendeteksi area yang cukup luas. Penelitian ini juga dapat digunakan untuk pencegahan kebakaran lahan gambut yang akhir – akhir ini sering terjadi.

1.2. Perumusan Masalah

Permasalahan yang dibahas dalam tugas akhir ini adalah:

1. Bagaimana cara mengetahui karakteristik suhu permukaan tanah ?
2. Bagaimana Pengontrolan Jumlah Air Penyiraman ?

1.3. Tujuan Penelitian

Penelitian pada tugas akhir ini bertujuan sebagai berikut:

1. Penggunaan kamera termal untuk memprediksi suhu permukaan tanah.
2. Kontrol *fuzzy logic* digunakan untuk menentukan lama bukaan *valve*

1.4. Batasan Masalah

Batasan masalah dari tugas akhir ini adalah sebagai berikut:

1. Perangkat yang digunakan hanya mampu mendeteksi suhu dari jarak kurang dari 7 m.
2. Valve yang digunakan merupakan jenis on/off.

1.5. Metodologi Penelitian

Langkah-langkah yang dikerjakan pada tugas akhir ini adalah sebagai

berikut:

1. Studi Literatur

Studi literatur berisi kegiatan pengumpulan dan pengkajian dasar teori yang terpercaya untuk menunjang penulisan tugas akhir ini. Literatur dapat bersumber dari paper, jurnal, artikel, buku, maupun website yang bertaraf nasional dan internasional, serta dari hasil konsultasi dengan dosen pembimbing.

2. Perancangan Sistem Lahan

Tahap ini merupakan tahap pencarian informasi mengenai konsep yang dibutuhkan untuk merancang alat ini, yang bisa didapatkan dari studi literatur dan bimbingan dosen pembimbing. Kemudian dilakukan pengumpulan alat dan bahan yang dibutuhkan.

3. Perancangan Perangkat Lunak

Pada tahap ini dilakukan perancangan dan pengembangan pemrograman. Hal ini dilakukan untuk mempermudah dalam pembuatan perangkat keras.

4. Perancangan Sistem Keseluruhan

Setelah melakukan riset dari referensi yang berkaitan dengan pengerjaan tugas akhir ini, langkah berikutnya adalah melaksanakan perancangan sistem yang akan digunakan dalam implementasi gabungan dari perancangan *hardware* dan perancangan *software*. Sistem pada *software* akan ditanamkan kontrol Fuzzy Logic dengan input berupa data sensor IR dan output semprotan air pada tanah. Sistem *hardware* berupa perancangan mekanis yang mampu mengoptimalkan kinerja alat.

5. Tahap Pengujian

Pengujian dilakukan bertahap. Pertama dilakukan pengujian pada masing-masing bagian. bagian agar pemecahan masalah lebih mudah dilakukan. *Software* diuji terlebih dahulu sebelum diimplementasikan ke *hardware*. Kemudian, dilanjutkan dengan pengujian *hardware* yang sudah tertanam *software* yang telah dibuat. Bagian-bagian tersebut meliputi sistem sensor Inframerah dan sistem *fuzzy logic*. Tahap pengujian dilakukan untuk menentukan kendala dari sistem yang telah dibuat, *software* maupun *hardware*.

6. Analisa dan Evaluasi

Analisa dilakukan terhadap hasil pengujian sehingga karakteristik *software* dan *hardware* dapat diketahui. Analisa dilakukan pada bagian

utama, yaitu sistem pengeluaran air, berapa lama *valve* akan dibuka untuk suatu keadaan tertentu. Apabila karakteristik dari bagian utama tersebut belum sesuai, maka perlu dilakukan evaluasi pada sistem untuk dirancang dan diuji kembali.

7. Penyusunan Laporan Tugas Akhir

Penyusunan laporan akhir berisi seluruh hal yang berkaitan dengan tugas akhir yang telah dikerjakan yaitu meliputi pendahuluan, studi literatur, tinjauan pustaka, perancangan dan pembuatan sistem, pengujian dan analisa, serta penutup.

1.6. Sistematika Penulisan

Dalam buku tugas akhir ini, pembahasan mengenai sistem yang dibuat terbagi menjadi lima bab dengan sistematika penulisan sebagai berikut:

- **BAB I : Pendahuluan**
Bab ini meliputi penjelasan latar belakang, rumusan masalah, batasan masalah, tujuan, metodologi, sistematika penulisan, dan relevansi.
- **BAB II : Tinjauan Pustaka**
Pada bab ini berisi mengenai teori yang mendasari penyusunan laporan tugas akhir secara umum khususnya yang berhubungan komponen yang akan digunakan.
- **BAB III : Perancangan Sistem**
Bab ini menjelaskan tentang perencanaan sistem yang meliputi perangkat keras (*hardware*) dan perangkat lunak (*software*) untuk pembuatan alat ini.
- **BAB IV : Pengujian dan Analisis**
Pada bab ini menguraikan tentang pengujian alat pada pesawat dan analisa hasil pengujian.
- **BAB V : Penutup**
Bab ini berisi tentang kesimpulan yang diperoleh dari pembuatan alat serta saran untuk pengembangan lebih lanjut.

1.7. Relevansi

Kontrol Kelembapan Tanah Menggunakan Kamera Termal dan *Fuzzy Logic* ini merupakan sebuah inovasi untuk membantu petani dalam menjaga kelembapan tanah tanpa harus menggunakan sensor yang ditanam, dimana dengan kamera termal jangkauan yang didapatkan bisa lebih luas, mengurangi pemakaian sensor yang berlebihan, mengurangi

kerusakan pada sensor karena kamera termal digunakan tanpa adanya kontak dengan tanah. Hal ini dapat mempermudah petani karena dapat memonitoring sekaligus mengontrol kelembapan tanah dengan efektif.

.....*Halaman ini sengaja dikosongkan*.....

BAB II

TINJAUAN PUSTAKA

Suatu penelitian memerlukan teori-teori yang sudah ada sebelumnya untuk dikaji lebih dalam memperkuat argumen penulis. Teori tersebut digunakan untuk membantu penulis dan sebagai dasar dalam membuat suatu penelitian.

Pada bab ini terdapat teori dasar yang menjadi landasan untuk merumuskan dan menyelesaikan masalah yang akan dibahas pada penelitian ini. Pada bagian ini terdapat tinjauan pustaka tentang komponen yang akan digunakan untuk membuat alat pada penelitian ini.

2.1. Tanah

Tanah adalah material yang terdiri dari beberapa bahan, yaitu mineral, bahan organik, organisme hidup, gas dan air [5]. Tanah menopang kehidupan tanaman dan mengandung banyak organisme hidup. Tanah, bersama dengan udara dan air adalah salah satu dari tiga sumber daya alam yang paling penting, yang kita tidak dapat hidup tanpanya. Tanah produktif mengandung sekitar 46% mineral, 4% bahan organik, 25% air dan 25% udara.



Gambar 2.1 Tanah Produktif [6].

2.1.1. Tekstur Tanah

Tanah terdiri dari ruang udara dan bahan organik, tetapi sebagian besar partikel mineral. Ada tiga jenis mineral tanah : tanah liat, lanau, dan pasir. Persentase relative dari masing – masing partikel ini dalam tanah menentukan teksturnya. Tekstur tanah tidak akan berubah kecuali anda

benar – benar menggali tanah dan menggantinya. Tanah liat memiliki partikel tanah terkecil (diameternya kurang dari 0.0001 inci) dan sedikitnya jumlah air dan ruang udara antar partikel. Akibatnya, tanah dengan tanah liat minimal 50 persen akan memiliki karakteristik yang berlawanan dengan pasir. Mengalir, mengering dan menghangat secara perlahan, tetapi sangat subur dan, sekali basah, menahan air dengan baik.



Gambar 2.2 Tanah Liat.

Ukuran partikel tanah lanau adalah antara tanah liat dan pasir (diameter 0.0001 hingga 0.01 inci). Karakteristik tanah berlumpur mirip dengan tanah liat, tetapi dimoderasi oleh ukuran partikelnya yang lebih besar. Pasir memiliki partikel tanah terbesar (diameter 0.01 – 0.4 inci) dan pori – pori terbesar di antara partikel. Tanah dengan 50 persen pasir cenderung mengalir dengan baik, dan mengering dan cepat panas, tetapi juga cenderung kurang subur dan tidak memiliki kelembapan yang baik sepanjang musim.



Gambar 2.3 Tanah Pasir [7].

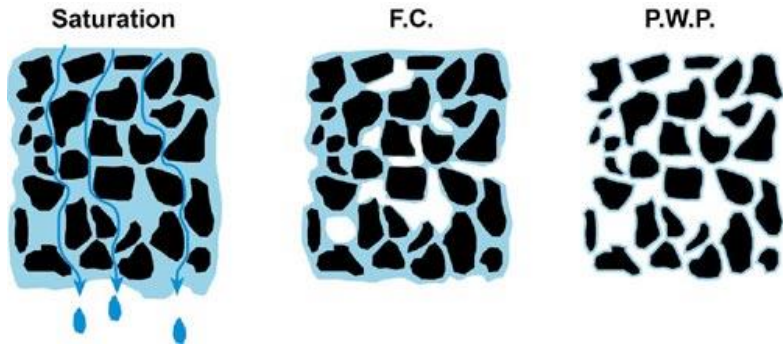
2.1.2. Struktur Tanah

Struktur tanah didefinisikan sebagai susunan 3-D dari partikel primer di tanah, yaitu bagaimana partikel primer terletak dan terhubung satu sama lain. Struktur tanah menggambarkan konfigurasi fisik tanah. Cara partikel pasir, lumpur, dan tanah liat dikelompokkan bersama disebut struktur tanah. Partikel pasir di tanah pasir disatukan bersama dengan lemah dan tidak membentuk agregat, sedangkan partikel tanah liat di tanah liat siap membentuk agregat. Struktur agregat adalah hasil dari berbagai proses di tanah yang dalam bentuk kombinasi agregat. Proses pembentukan struktur ini mempengaruhi struktur tanah dalam interaksi dinamis dengan pengolahan tanah.

Ketika partikel – partikel tanah liat terikat Bersama dalam agregat, sifat – sifat tanah hampir selalu diperbaiki. Efek pentingnya adalah pengolahan tanah menjadi lebih mudah. Pergerakan udara juga menjadi lebih baik, meningkatkan pengangkutan oksigen ke dan karbon dioksida dari sistem akar. Struktur agregat juga meningkatkan permeabilitas tanah terhadap air dan kapasitas penampungan airnya. Struktur tanah yang baik membuat tanah liat yang subur menjadi tempat tumbuh yang lebih produktif, karena semua fungsi penting untuk tanaman dan system akarnya disediakan. Ini memberi pertumbuhan lebih cepat dan hasil lebih tinggi [8].

2.1.3. Klasifikasi Kelembapan Tanah

Dalam pengelolaan air tanah, klasifikasi dilakukan berdasarkan ketersediaan air untuk tanaman di tanah. Jika tanah berada pada kapasitas retensi maksimumnya, maka tanah tersebut masuk dalam kategori saturasi (jenuh). Pada titik ini semua pori-pori tanah terisi air. Kejenuhan biasanya terjadi untuk periode waktu yang singkat, baik pada saat hujan deras atau ketika tanah diairi. Setelah sepenuhnya dibasahi dan semua air gravitasi telah terkuras habis, tanah masuk dalam kategori kapasitas lapangan (Field Capacity/FC). Pada kapasitas lapangan, tanah menampung jumlah air maksimum yang dapat disimpan dan digunakan oleh tanaman. Selain itu, ada cukup ruang pori berisi udara untuk memungkinkan aerasi. Jika tanah tampak berdebu dan kering, maka tanah masuk dalam kategori *Permanent Wilting Point* (PWP). PWP didefinisikan sebagai kelembaban tanah minimum di mana tanaman layu dan tidak bisa lagi memulihkan turgiditasnya. Namun, air masih ada di mikropori dan dalam film yang sangat tipis di sekitar partikel tanah, tetapi air ini tidak bisa digunakan oleh akar tanaman dan mikroba.



Gambar 2.4 Klasifikasi Kelembapan Tanah [9].

2.2. Radiasi Benda Hitam dan Spektrum Inframerah

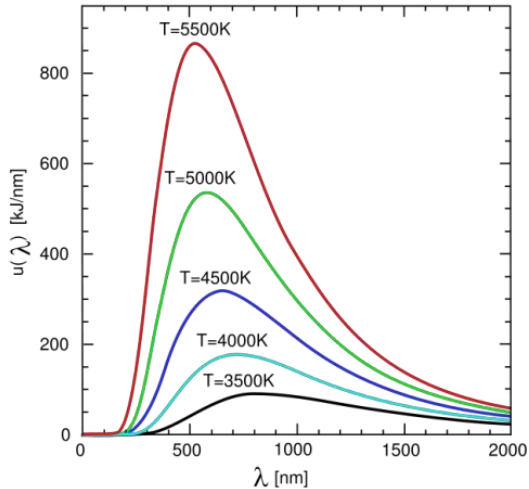
Radiasi benda hitam mengacu pada objek atau system yang menyerap semua insiden radiasi di atasnya dan memancarkan kembali energi yang merupakan karakteristik dari system radiasi ini saja, tidak tergantung pada jenis radiasi yang terjadi di atasnya. Energi yang terpencar dapat dianggap dihasilkan oleh gelombang berdiri atau mode resonansi dari rongga yang memancar [10]. Suhu menentukan distribusi Panjang gelombang radiasi elektromagnetik. Misalnya cat berwarna putih sangat reflektif terhadap cahaya tampak, dan tampak putih pada mata manusia karena memantulkan sinar matahari, yang memiliki Panjang gelombang puncak sekitar 0.5 mikrometer. Namun, emisivitasnya pada suhu sekitar -5°C (23°F), Panjang gelombang puncak sekitar 12 mikrometer, adalah 0.95. jadi untuk radiasi termal tampak hitam [11].

Benda hitam, yang merupakan emitor “ideal” atau “sempurna”(yang berarti sifat emisinya tidak bervariasi berdasarkan lokasi atau komposisi objek), memancarkan spektrum cahaya dengan sifat sebagai berikut :

Pertama, semakin panas benda hitam, semakin banyak cahaya yang dikeluarkan di semua panjang gelombang. Jika kita membandingkan dua benda hitam, terlepas dari berapa panjang gelombang cahaya yang kita amati, benda hitam yang lebih panas akan mengeluarkan lebih banyak cahaya daripada yang lebih dingin.

Kedua, spektrum dari sebuah benda hitam adalah kontinu, yaitu memancarkan cahaya pada semua panjang gelombang dan memiliki puncak pada panjang gelombang tertentu. Puncak kurva benda hitam dalam spektrum bergerak ke panjang gelombang yang lebih pendek untuk

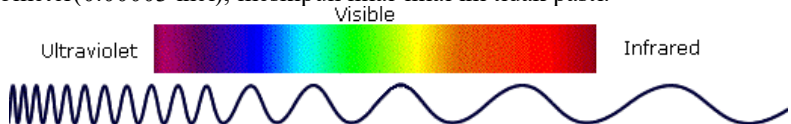
objek yang lebih panas.



Gambar 2.5 Kurva Radiasi Benda Hitam [12].

Gelombang inframerah, atau cahaya inframerah, adalah bagian dari spektrum elektromagnetik. Gelombang ini biasa ditemui di kehidupan sehari-hari; mata manusia tidak dapat melihatnya, tetapi manusia dapat mendeteksinya sebagai panas [13]. Gelombang inframerah dapat dibagi menjadi beberapa wilayah spektrum, atau band, berdasarkan panjang gelombang; Namun tidak ada definisi yang seragam dari batas-batas yang tepat setiap band [14].

Gelombang inframerah lebih panjang daripada cahaya tampak, tepat di luar ujung merah spektrum yang terlihat. Inframerah berada di kisaran spektrum elektromagnetik antara gelombang mikro dan cahaya tampak. Inframerah memiliki frekuensi dari sekitar 3GHz hingga sekitar 400 THz dan panjang gelombang sekitar 30 sentimeter(12 inci) hingga 740 nanometer(0.00003 inci), meskipun nilai-nilai ini tidak pasti.



Gambar 2.6 Spektrum Gelombang Inframerah [15].

2.3. Kamera Termal

Kamera termal adalah perangkat yang menerjemahkan energi panas menjadi cahaya tampak untuk menganalisa objek atau pemandangan tertentu. Citra yang dihasilkan dikenal sebagai termogram dan dianalisis melalui proses yang disebut termografi. Kamera pencitraan termal adalah perangkat canggih yang memproses gambar yang diambil dan menampilkannya di layar. Gambar – gambar ini dapat digunakan untuk diagnosis secara langsung atau diproses melalui perangkat lunak khusus untuk evaluasi lebih lanjut, keakuratan, dan keluaran laporan. Kamera pencitraan termal mengukur suhu ke tingkat berikutnya; alih – alih mendapatkan nilai dari suhu, hasil yang didapat adalah gambar yang menunjukkan perbedaan suhu permukaan [16].

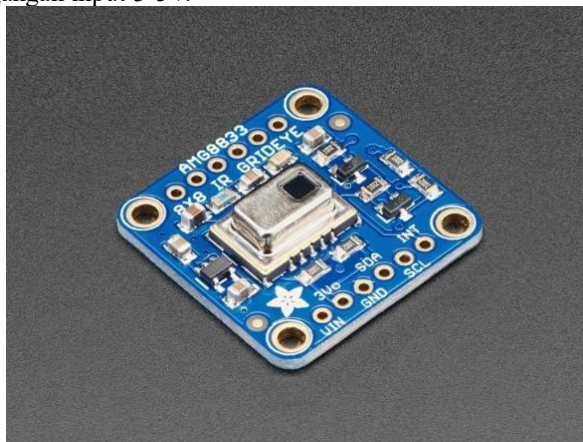
Cahaya yang terlihat sangat berguna untuk manusia dan makhluk lain, tetapi itu hanya sebagian kecil dari spektrum elektromagnetik. Radiasi inframerah, jenis yang dihasilkan oleh panas, membutuhkan lebih banyak “ruang” dalam spektrum. Interaksi panas yang diserap, dipantulkan, dan kadang-kadang bahkan ditransmisikan adalah apa yang dipotret dan dievaluasi oleh kamera termal. Tingkat radiasi termal yang dipancarkan oleh suatu objek disebut tanda panasnya. Semakin panas objek yang diberikan, semakin banyak radiasi akan menyebar ke lingkungan. Kamera termal mampu membedakan sumber panas dan perbedaan yang sangat kecil dalam radiasi panas. Kamera termal mengkompilasi data ini menjadi “peta panas” lengkap dibedakan oleh tingkat panas.



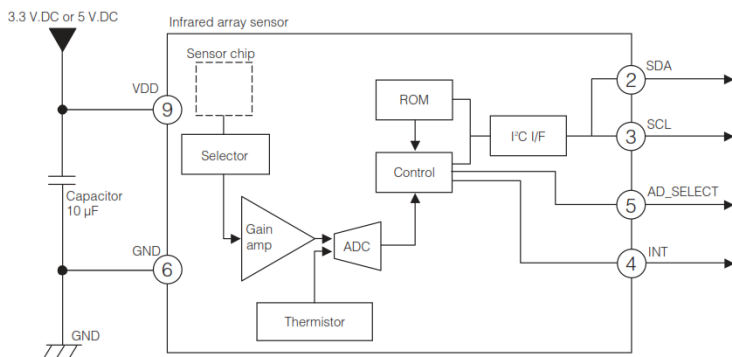
Gambar 2.7 Kamera Termal [17].

2.4. Infrared Array Sensor Grid-EYE (AMG8833)

Sensor ini merupakan sensor inframerah atau termal dengan array 8x8 (64 piksel). Ketika terhubung ke mikrokontroler, ia akan menampilkan array dari 64 pembacaan suhu inframerah. Sensor ini akan mengukur suhu mulai dari 0⁰ C hingga 80⁰ C (32⁰ F hingga 176⁰ F) dengan akurasi ±2.5⁰ C (4.5⁰ F), dapat mendeteksi manusia hingga jarak 7 meter (23) kaki, dengan frame rate maksimum 10Hz. AMG8833 beroperasi dengan tegangan input 3-5V.



Gambar 2.8 Sensor AMG8833



Gambar 2.9 Rangkaian Internal Sensor AMG8833 [18].

2.5. Arduino

Tabel 2.1. Spesifikasi Arduino Uno

Mikrokontroler	ATmega328
Tegangan operasi	5V
Tegangan <i>input</i> (rekomendasi)	7 – 12V
Tegangan <i>input</i> (limit)	6 – 20V
Pin I/O digital	14 (6 diantaranya menyediakan keluaran PWM)
Pin <i>input</i> 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 <i>bootloader</i>
SRAM	2 KB
EEPROM	1 KB
Kecepatan <i>clock</i>	16 MHz

Arduino Uno merupakan salah satu mikrokontroler *single-board* berbasis ATmega328 yang bersifat yang dapat memudahkan pengguna untuk mengetahui dan mengembangkan cara kerja perangkat tersebut. Perangkat tersebut memiliki 14 *input/output* digital yang mana pin 3, 5, 6, 9, 10, dan 11 dapat digunakan sebagai PWM *output*, enam *input* analog, pin *power* (VIN,5V,3V3, dan GND), resonator keramik 16 MHz, USB *connector*, *power jack*, ICSP *header*, dan *reset button*. Sebagai contoh penggunaan Arduino Uno adalah sistem kontrol kekeruhan akuarium, pembacaan meteran air dimana Arduino Uno digunakan sebagai prosesor data dari node sensor, dan pembacaan frekuensi pada deteksi jalur pipa terpendam [20].

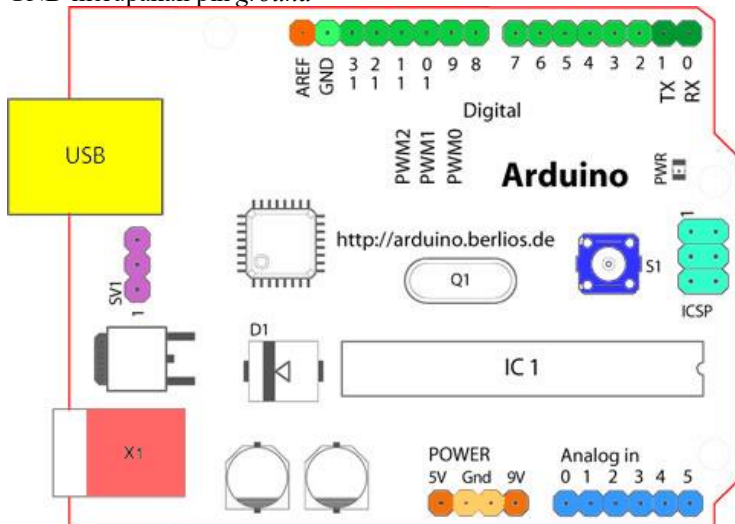


Gambar 2.10 Arduino UNO [19].

2.5.1. Pin Power

Sumber tegangan yang diberikan ke Arduino Uno dapat berasal dari hubungan USB atau *external power supply* seperti *AC-to-DC adapter* atau baterai. Besar tegangan eksternal yang digunakan untuk pengoperasian yaitu antara 6-20 V. Ketika tegangan eksternal yang diberikan di bawah 7 V, pin 5 V menghasilkan tegangan di bawah 5 V karena Arduino Uno tidak stabil. Sedangkan ketika diberi tegangan eksternal lebih dari 12 V, *voltage regulator* milik Arduino akan terlalu panas dan Arduino berkemungkinan bekerja tidak stabil. Sehingga direkomendasikan tegangan eksternal yang untuk mengoperasikan Arduino Uno adalah 7-12 V. 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*



Gambar 2.11 Arduino Uno [21].

2.5.2. Input dan Output

Pada Arduino Uno terdapat pin digital yang digunakan sebagai masukan atau keluaran dengan jumlah pin 14 buah. Setiap pin digital tersebut bekerja pada tegangan 5V dan arus maksimum 40 mA. Penggunaan pin digital dapat diakses dengan fungsi `pinMode ()`, `digitalWrite ()`, dan `digitalRead ()`. Selain itu, beberapa pin tersebut memiliki fungsi khusus:

- Pin 0(RX) dan 1(TX) digunakan sebagai serial komunikasi dimana pin 0 digunakan untuk menerima data serial TTL sedangkan pin 1 digunakan untuk mengirimkan data serial TTL.
- Pin 2 dan 3 digunakan sebagai *external interrupt*. Pin ini dapat dikonfigurasi untuk men-*trigger interrupt* ketika nilai rendah, *rising edge* atau *falling edge*, atau ketika terdapat 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.

Arduino Uno juga memiliki 6 pin analog yang berfungsi sebagai masukan analog. Pin tersebut diberi label A0 hingga A5. Nilai tegangan keluaran dari pin analog bernilai mulai dari 0 - 5 V dan menyediakan resolusi 10 bit. Pin A4 dan A5 membantu komunikasi TWI dalam Arduino Uno dimana A4 sebagai SDA dan A5 sebagai SCL. Selain itu terdapat pin AREF yang berfungsi sebagai tegangan referensi untuk masukan analog dan pin RESET digunakan untuk menambah tombol reset [22].

2.5.3. 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 Uno 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 [23].

2.5.4. Pemrograman

Arduino Uno dapat di program dengan Arduino Software. Sebelum meng-upload program ke Arduino Uno, terlebih dahulu pilih Arduino Uno pada menu Tool > Board. ATmega328 pada Arduino Uno telah dilengkapi dengan bootloader yang memungkinkan peng-upload-an program baru ke Arduino Uno tanpa menggunakan pemrograman external hardware.

Setelah membuat rangkaian yang diinginkan baik di breadboard ataupun yang sudah 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 nsatu sketsa pada suatu waktu. Perangkat lunak yang digunakan untuk membuat sketsa Arduino disebut IDE yang merupakan singkatan dari *Integrated Development Environment* [24].

2.6. Fuzzy Logic

Fuzzy Logic telah ditemukan dan digunakan untuk menggambarkan sistem yang tidak pasti sejak tahun 60-an pada abad ke-20. Teknik ini juga telah digunakan dalam sistem kontrol. Kontrol *Fuzzy* sekarang merupakan metode kontrol standar dari sebagian besar sistem industri [25].

Fuzzy Logic adalah generalisasi logika standar, di mana konsep dapat memiliki tingkat kebenaran dimana saja antara 0 dan 1. *Fuzzy Logic* adalah teknik untuk merepresentasikan dan memanipulasi informasi yang tidak pasti. Dalam logika proporsional yang lebih tradisional, setiap fakta atau proposisi, seperti “hujan akan turun besok”, harus benar atau salah. Namun banyak informasi yang digunakan orang tentang dunia melibatkan beberapa tingkat ketidakpastian, seperti teori probabilitas. *Fuzzy Logic* melampirkan nilai antara 0 dan 1 untuk setiap proposisi untuk mewakili ketidakpastian, sedangkan teori probabilitas mengukur derajat ke yang proposisinya benar [26].

Pada *Fuzzy Logic*, ada beberapa tahapan, yaitu *Fuzzyfikasi*, *Rule Base*, dan *Defuzzyfikasi*. *Fuzzyfikasi* adalah proses perubahan nilai numerik menjadi linguistik, seperti panas, dan dingin. *Rule Based* adalah proses pngkonversian input fuzzy menggunakan aturan “if-then” menjadi output fuzzy [27].

2.6.1. Arsitektur Fuzzy Logic

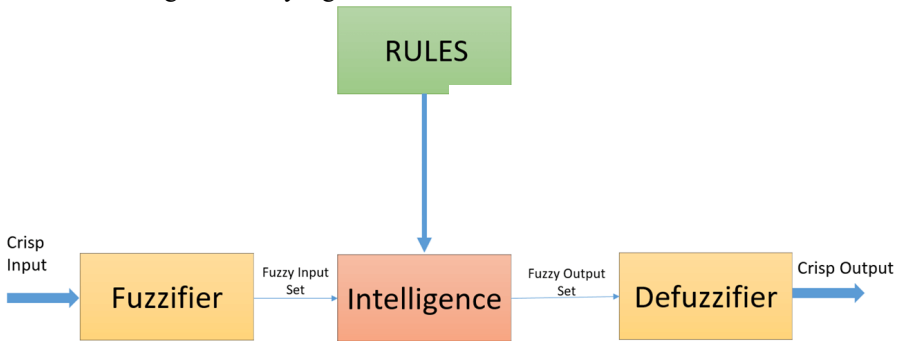
Arsitektur *Fuzzy Logic* memiliki 4 bagian utama, yang pertama adalah *Rule Base*. *Rule Base* berisi semua aturan dan kondisi “if-then” yang diberikan untuk mengontrol sistem pengambilan keputusan. Pembaruan terbaru dalam teori *fuzzy* menyediakan metode yang bervariasi untuk desain dan *tuning* dari controller *fuzzy*. Pembaruan ini

secara signifikan mengurangi jumlah himpunan aturan fuzzy.

Bagian selanjutnya adalah Fuzzyfikasi, dimana pada tahap ini dilakukan konversi terhadap input. Nilai dari input diubah menjadi himpunan *fuzzy*. Input diukur oleh sensor dan masuk ke sistem kontrol untuk proses yang lebih jauh, seperti suhu permukaan, tekanan, dan sebagainya.

Kemudian ada *Inference Engine* yang berguna untuk mengukur tingkat kecocokan antara input *fuzzy* dan aturan-aturan yang telah dibuat. Berdasarkan persentase kecocokan, ini akan menentukan aturan mana yang perlu diterapkan berdasarkan input yang telah diberikan. Setelah ini, aturan yang ditetapkan dikombinasikan untuk mengembangkan kontrol.

Bagian terakhir adalah Defuzzyfikasi, dimana pada tahap ini dilakukan konversi terhadap himpunan *fuzzy* menjadi sebuah nilai. Ada banyak tipe dari teknik yang tersedia, jadi kita harus memilih yang paling cocok dengan sistem yang telah kita buat.



Gambar 2.12 Arsitektur Fuzzy [28].

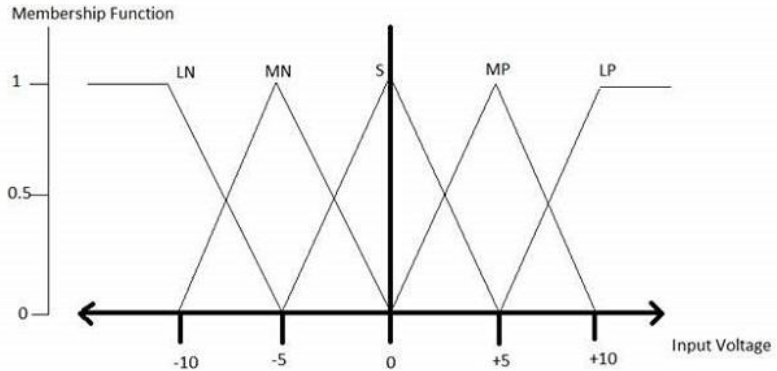
2.6.2. Fungsi Keanggotaan

Fungsi Membership memungkinkan kita untuk mengukur istilah linguistik dan membuat himpunan *fuzzy* ke dalam bentuk grafik. Fungsi membership untuk sebuah himpunan *fuzzy* A pada himpunan X didefinisikan sebagai : $\mu_A : X \rightarrow [0,1]$.

Disini, setiap elemen dari X dipetakan menjadi sebuah nilai di antara 0 dan 1. Ini disebut nilai keanggotaan atau tingkat keanggotaan. Ini mengukur tingkat keanggotaan elemen pada X ke himpunan fuzzy A.

- Sumbu x mewakili himpunan.
- Sumbu y mewakili tingkat keanggotaan pada interval $[0,1]$.

Ada banyak fungsi keanggotaan yang dapat diaplikasikan untuk membuat nilai numerik menjadi himpunan fuzzy. Fungsi keanggotaan sederhana digunakan karena penggunaan fungsi kompleks tidak menambah presisi dalam output.



Gambar 2.13 Contoh Fungsi Keanggotaan [29].

2.7. LCD (Liquid Crystal Display)

2.7.1. Dasar Teori LCD

LCD merupakan salah satu komponen elektronika yang berfungsi sebagai tampilan suatu data, baik karakter, huruf, atau grafik. LCD membutuhkan tegangan dan daya yang kecil sehingga sering digunakan untuk aplikasi pada kalkulator, arloji digital, dan instrumen elektronik seperti multimeter digital. LCD memanfaatkan silikon dan galium dalam bentuk kristal cair sebagai pemancar cahaya. Pada layar LCD, setiap matrik adalah susunan dua dimensi piksel yang dibagi dalam baris dan kolom. Dengan demikian, setiap pertemuan baris dan kolom terdiri dari LED pada bidang latar (*backplane*), yang merupakan lempengan kaca bagian belakang dengan sisi dalam yang ditutupi oleh lapisan elektroda transparan. Dalam keadaan normal, cairan yang digunakan memiliki warna cerah. Kemudian daerah-daerah tertentu pada cairan tersebut warnanya akan berubah menjadi hitam ketika tegangan diterapkan antara bidang latar dan pola elektroda yang terdapat pada sisi dalam kaca bagian depan. Keunggulan menggunakan LCD adalah konsumsi daya yang relatif kecil dan menarik arus yang kecil (beberapa mikro ampere), sehingga alat atau sistem menjadi portable karena dapat menggunakan catu daya yang kecil. Keunggulan lainnya adalah ukuran LCD yang pas yakni tidak terlalu kecil dan tidak terlalu besar kemudian tampilan yang

diperlihatkan dari LCD dapat dibaca dengan mudah dan jelas [30].



Gambar 2.14 LCD 16x2 [31].

Spesifikasi pada LCD 16x2 adalah sebagai berikut :

- Terdiri dari 16 kolom dan 2 baris
- Mempunyai 192 karakter yang tersimpan
- Tegangan kerja 5V
- Memiliki ukuran yang praktis

2.7.2. Prinsip Kerja LCD 16x2

Prinsip kerja LCD 16x2 adalah dengan menggunakan lapisan film yang berisi kristal cair dan diletakkan di antara dua lempeng kaca yang telah dipasang elektroda logam transparan. Saat tegangan dicatukan pada beberapa pasang elektroda, molekul-molekul kristal cair akan menyusun agar cahaya yang mengenainya akan diserap. Dari hasil penyerapan cahaya tersebut akan terbentuk huruf, angka, atau gambar sesuai bagian yang diaktifkan. Untuk membentuk karakter atau gambar pada kolom dan baris secara bersamaan digunakan metode *screening*. Metode *screening* adalah mengaktifkan daerah perpotongan suatu kolom dan baris secara bergantian dan cepat sehingga seolah-olah aktif semua [30].

2.7.3. Deskripsi Pin LCD 16x2

Berikut ini adalah tabel deskripsi pin pada LCD 16x2:

Tabel 2.2. Deskripsi Pin pada LCD 16x2

Pin	Simbol	I/O	Deskripsi
1	VSS	--	Ground
2	VCC	--	+ 5 V power suplay
3	VEE	--	Power suplay source to control contrast

4	RS	I	Register select: RS = 0 to select instruksi. Command register; RS =1 to select data reg.
5	R/W	I	Read/Write: R/W =0 for write, R/W= 1 for read
6	E	I	Enable
7	DB0	I/O	The 8-bit data bus
8	DB1	I/O	The 8-bit data bus
9	DB2	I/O	The 8-bit data bus
10	DB3	I/O	The 8-bit data bus
11	DB4	I/O	The 8-bit data bus
12	DB5	I/O	The 8-bit data bus
13	DB6	I/O	The 8-bit data bus
14	DB7	I/O	The 8-bit data bus

2.8. I2C

Inter Integrated Circuit atau yang lebih dikenal dengan sebutan I2C adalah merupakan standar komunikasi serial dua arah dengan menggunakan dua buah saluran yang didesain khusus untuk pengontrolan IC tersebut. Secara garis besar sistem I2C itu sendiri tersusun atas dua saluran utama yaitu, saluran SCL (*serial clock*) dan SDA (*serial data*) yang membawa informasi data antara I2C dengan sistem pengontrolnya [32].



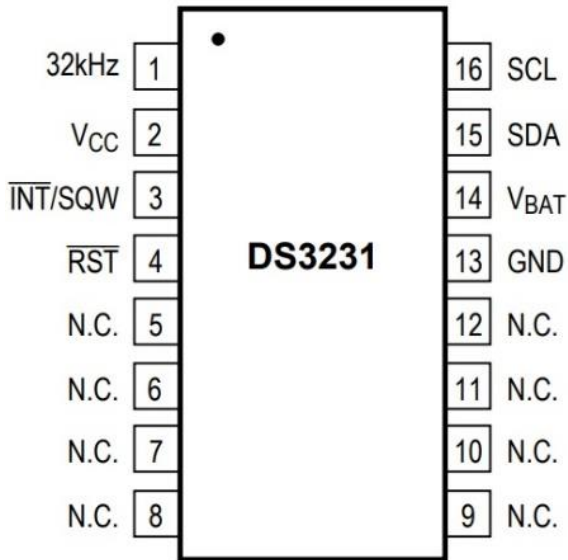
Gambar 2.15 LCD modul I2C [33].

Perangkat yang dihubungkan dengan I2C ini dapat difungsikan sebagai *master* atau *slave*. *Master* adalah perangkat yang memulai *transfer* pada data dengan membentuk sinyal *stop*, dan membangkitkan sinyal *clock*. Sedangkan *slave* adalah perangkat yang telah diberikan alamat oleh *master*.

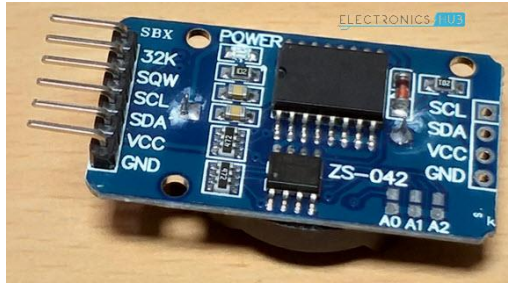
Berikut ini merupakan beberapa kondisi ketika melakukan proses *transfer* data pada I2C bus, yaitu *transfer* data hanya dapat dilakukan ketika bus tidak dalam keadaan sibuk, lalu selama proses *transfer* data keadaan pada pin SDA haruslah stabil selama pin SCL dalam keadaan tinggi.

2.9. RTC DS3231

Real Time Clock atau RTC adalah sebuah perangkat untuk mengetahui waktu yang dibuat dalam bentuk rangkaian terintegrasi atau (IC). DS3231 adalah sebuah RTC yang dikembangkan oleh Maxim Integrated. Harganya tidak terlalu mahal, tetapi hampir akurat dengan system komunikasi menggunakan antarmuka I2C. Fitur yang menarik dari DS3231 ini adalah DS3231 ini sudah mempunyai *Crystal Oscillator* dan sensor suhu sehingga tidak perlu lagi dihubungkan ke Crystal Eksternal.



Gambar 2.16 Pin Diagram dari DS3231.



Gambar 2.17 Modul RTC DS3231.

Tabel 2.3. Deskripsi Pin pada RTC DS3231 [34].

Nomor Pin	Nama Pin	Deskripsi
1	32KHz	32KHz Output
2	Vcc	DC Power Pin
3	INT/SQW	Active Low Interrupt or Square wave Output
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

2.10. HC-05

HC-05 adalah sebuah Bluetooth SPP (Serial Port Protocol) yang mudah digunakan untuk komunikasi serial wireless (nirkabel) yang mengkonversi port serial ke Bluetooth. HC-05 menggunakan modulasi Bluetooth V2.0 + EDR (Enhanced Data Rate) 3 Mbps dengan memanfaatkan gelombang radio berfrekuensi 2,4 GHz.

Modul ini dapat digunakan sebagai slave maupun master. HC-05 memiliki 2 mode konfigurasi, yaitu AT mode dan *Communication* mode. AT mode berfungsi untuk melakukan pengaturan konfigurasi dari HC-05. Sedangkan *Communication* mode berfungsi untuk melakukan komunikasi Bluetooth dengan piranti lain.

Dalam penggunaannya, HC-05 dapat beroperasi tanpa menggunakan driver khusus. Untuk berkomunikasi antar Bluetooth, minimal harus memenuhi dua kondisi berikut :

- Komunikasi harus antara master dan slave
- Password harus benar (saat melakukan pairing)

Adapun spesifikasi dari HC-05 adalah :

Hardware :

- Sensitivitas -80dBm (Typical)
- Daya transmit RF sampai dengan +4dBm.
- Operasi daya rendah 1,8V – 3,6V I/O.
- Kontrol PIO.
- Antarmuka UART dengan baudrate yang dapat diprogram.
- Dengan antenna terintegrasi.

Software :

- Default baudrate 9600, Data bit : 8, Stop bit = 1, Parity : No Parity, Mendukung baudrate : 9600, 19200, 38400, 57600, 115200, 230400 dan 460800.
- Auto koneksi pada saat device dinyalakan (default).
- Auto reconnect pada menit ke 30 ketika hubungan putus karena range koneksi [35].



Gambar 2.18 Modul Bluetooth HC-05 [36].

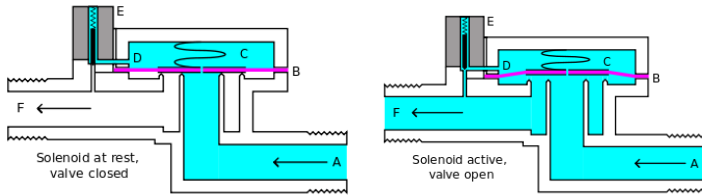
2.11. Solenoid Valve

Solenoid Valve adalah katup yang digerakan oleh energi listrik, mempunyai koil sebagai penggeraknya yang berfungsi untuk menggerakkan piston yang dapat digerakkan oleh arus AC maupun DC, Solenoid Valve mempunyai lubang keluaran, lubang masukan dan lubang exhaust.

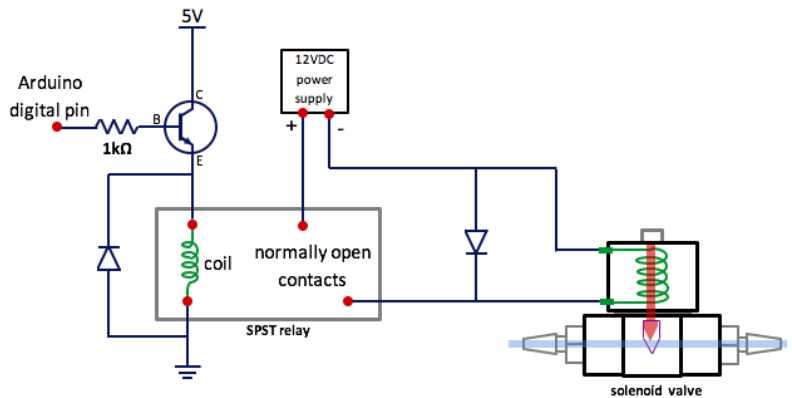


Gambar 2.19 Solenoid Valve [37].

Solenoid Valve merupakan katup listrik yang mempunyai koil sebagai penggerak yang mana ketika koil mendapat supply tegangan maka koil tersebut akan berubah menjadi medan magnet sehingga menggerakkan piston pada bagian dalamnya ketika piston berpindah posisi maka valve akan terbuka.



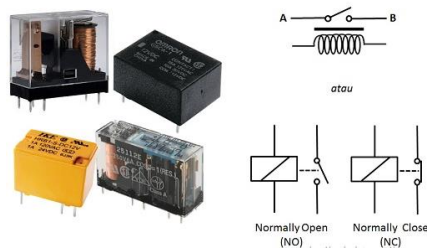
Gambar 2.20 Cara Kerja Solenoid Valve [38].



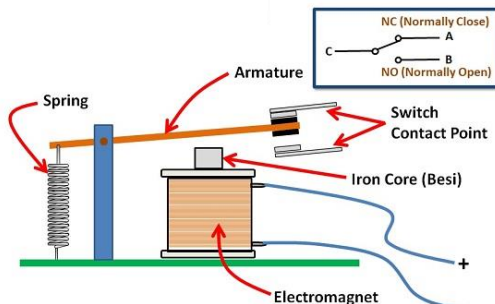
Gambar 2.21 Skematik Solenoid Valve [39].

2.12. Relay

Relay merupakan komponen elektronika berupa saklar atau switch elektrik yang dioperasikan secara listrik dan terdiri dari 2 bagian utama yaitu electromagnet (coil) dan mekanikal (seperangkat kontak Saklar/Switch). Komponen elektronika ini menggunakan prinsip elektromagnetik untuk menggerakkan saklar sehingga dengan arus listrik yang kecil (low power) dapat menghantarkan listrik yang bertegangan lebih tinggi.



Gambar 2.22 Bentuk dan Simbol Relay [40].



Gambar 2.23 Struktur Sederhana Relay

Kontak Poin (Contact Point) Relay terdiri dari 2 jenis yaitu :

- Normally Close (NC) yaitu kondisi awal sebelum diaktifkan akan selalu berada di posisi CLOSE (tertutup)
- Normally Open (NO) yaitu kondisi awal sebelum diaktifkan akan selalu berada di posisi OPEN (terbuka)

Berdasarkan gambar diatas, sebuah Besi (Iron Core) yang dililit oleh sebuah kumparan Coil yang berfungsi untuk mengendalikan Besi tersebut. Apabila Kumparan Coil diberikan arus listrik, maka akan timbul gaya Elektromagnet yang kemudian menarik Armature untuk berpindah

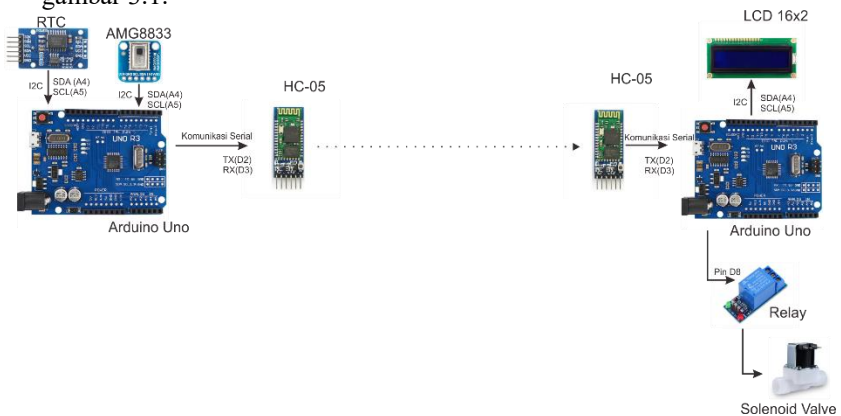
dari Posisi sebelumnya (NC) ke posisi baru (NO) sehingga menjadi Saklar yang dapat menghantarkan arus listrik di posisi barunya (NO). Posisi dimana Armature tersebut berada sebelumnya (NC) akan menjadi OPEN atau tidak terhubung. Pada saat tidak dialiri arus listrik, Armature akan kembali lagi ke posisi Awal (NC). Coil yang digunakan oleh Relay untuk menarik Contact Poin ke Posisi Close pada umumnya hanya membutuhkan arus listrik yang relatif kecil [41].

.....*Halaman ini sengaja dikosongkan*.....

BAB III PERANCANGAN SISTEM

Pada bab ini dijelaskan perancangan sistem secara keseluruhan. Alat yang dirancang bertujuan untuk membuat pengontrol kelembapan tanah menggunakan sensor kamera termal dan *Fuzzy Logic* sebagai kontrolnya sehingga dapat memonitoring serta lebih mudah dalam mengontrol kelembapan tanah. Perancangan alat berupa sebuah Kontroller yang didalamnya dilengkapi dengan Arduino Uno sebagai mikrokontroler, sensor AMG8833, RTC DS3231, HC-05 Bluetooth Module, LCD 16x2, Relay, dan Solenoid Valve

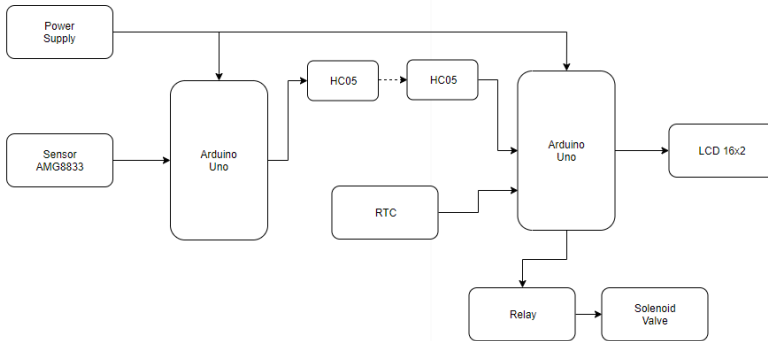
Arduino Uno berfungsi sebagai mikrokontroler yang mengatur kinerja dari controller, sensor AMG8833 digunakan untuk mendeteksi suhu permukaan tanah. Skema keseluruhan sistem dapat dilihat pada gambar 3.1.



Gambar 3.1. Skema Sistem Keseluruhan.

3.1. Diagram Blok Sistem

Sistem pada Kontroller ini terdiri dari perangkat keras dan perangkat lunak. Perangkat keras yang digunakan dalam sistem yaitu Arduino Uno, Sensor AMG8833. Untuk perangkat lunak yang digunakan yaitu program hasil pembacaan sensor AMG8833, program kontrol suhu menggunakan *fuzzy logic*. Diagram blok dari sistem tersebut dapat dilihat pada Gambar 3.2.



Gambar 3.2. Diagram Blok Sistem.

Bagian-bagian pada blok diagram terdiri dari:

- *Power Supply*
Daya yang dibutuhkan sistem dihasilkan oleh baterai Li-Po 12VDC/2A.
- Arduino Uno
Arduino Uno berperan sebagai perangkat keras yang menerima masukan dari sensor dan mengontrol aktuator yang akan digunakan.
- Sensor AMG8833
Sensor AMG8833 berguna untuk mendeteksi suhu permukaan tanah.
- HC05
HC05 digunakan untuk komunikasi *wireless* antara arduino.
- RTC (Real Time Clock)
RTC digunakan sebagai sensor untuk mendeteksi waktu pada kontroller agar dapat lebih mudah dalam menetapkan dan mengontrol bukaan valve.
- LCD 16x2
LCD 16xd digunakan sebagai antarmuka dari sistem yang dibuat. LCD menampilkan aksi yang akan dilakukan setelah pembacaan sensor.
- Relay
Relay berfungsi sebagai switch nyala dan tidaknya solenoid valve
- Solenoid Valve
Solenoid Valve berfungsi untuk membuka saluran air saat on, dan menutup saat off.

3.2. Perancangan Perangkat Keras

Perangkat keras pada tugas akhir meliputi desain tandon, koneksi Arduino terhadap setiap komponen yaitu, sensor AMG8833, HC-05, Real Time Clock, Relay dan *Solenoid Valve*.

3.2.1. Desain Tandon

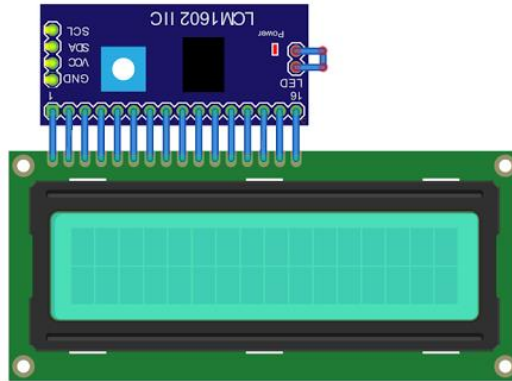
Desain tandon ini memiliki bentuk yang cukup sederhana. Kerangka desain utama berbentuk silinder. Pada bagian bawah depan tandon akan diletakkan solenoid valve untuk mengalirkan air.



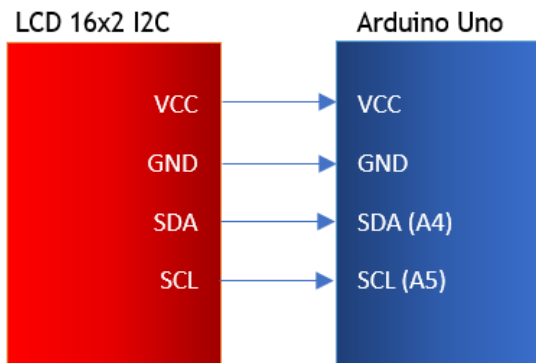
Gambar 3.3 Desain Tandon

3.2.2. LCD dengan I2C

LCD dipergunakan sebagai antarmuka pada alat ini, sehingga dapat membantu dan mempermudah dalam memberitahukan hasil pembacaan sensor. LCD 16x2 memiliki *pin out* sebanyak 16 buah, oleh karena itu agar dapat mengurangi jumlah pin yang dihubungkan dengan Arduino, penulis menggunakan LCM1602 I2C sebagai penghubung diantara keduanya. Antarmuka antara LCD 16x2 dengan I2C dapat dilihat pada Gambar 3.4, sedangkan antarmuka antara I2C dengan Arduino Uno dapat dilihat pada Gambar 3.5.



Gambar 3.4 LCD 16x2 terhadap I2C



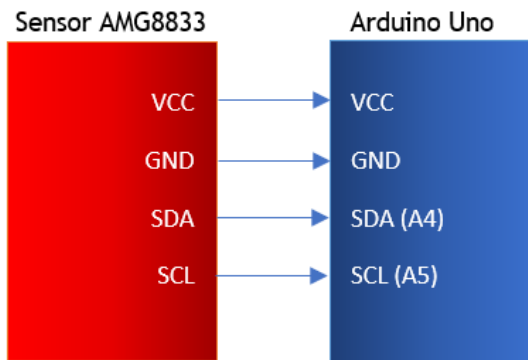
Gambar 3.5 Antarmuka LCD 16x2 I2C dengan Arduino Uno

3.2.3. Pembacaan sensor AMG8833

Sensor AMG8833 digunakan untuk mendeteksi suhu permukaan tanah, sebagai pengganti dari sensor kelembapan yang ditancapkan ke tanah. Jika sensor mendeteksi kekeringan pada tanah, maka mikrokontroller akan memproses hasil dari pembacaan sensor, kemudian menghidupkan valve untuk membasahi tanah yang kering.

Tabel 3.1. Spesifikasi Sensor AMG8833

Tegangan Masukan	3-5 VDC
Jumlah Piksel	64 (Vertikal 8 x Horizontal 8 Matriks)
Akurasi	+/- 2.5°C
Komunikasi	I2C
Suhu Maksimum Saat Beroperasi	+80°C
Suhu Minimum Saat Beroperasi	0°
Tipe Output	Digital



Gambar 3.6 Antarmuka Sensor AMG8833 dengan Arduino Uno

3.2.4. HC-05

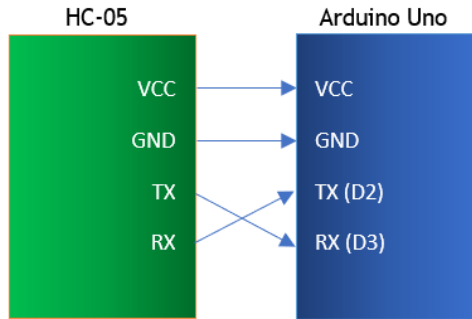
HC-05 adalah sebuah modul Bluetooth SPP (*Serial Port Protocol*) yang mudah digunakan untuk komunikasi serial *wireless* (nirkabel) yang mengkonversi port serial ke Bluetooth. Melalui HC-05 ini akan dilakukan komunikasi *wireless* antar arduino untuk menghidup/matikan *Solenoid Valve* yang berguna untuk mengontrol keluaran air.

Perintah untuk menghidup/matikan *Solenoid Valve* akan diproses oleh mikrokontroller setelah memastikan nilai dari pembacaan sensor. Jika hasil pembacaan sensor menunjukkan bahwa tanah berada pada kondisi yang kering, maka mikrokontroller akan mengirimkan informasi untuk menghidupkan *Solenoid Valve* melalui HC-05 yang ada pada rangkaian sensor, kepada HC-05 yang ada pada rangkaian aktuator. Jika informasi diterima dengan baik, maka mikrokontroller yang ada pada rangkaian aktuator akan mengaktifkan relay, sehingga mengaktifkan

Solenoid Valve. Penggunaan HC-05 bertujuan untuk mengurangi jumlah kabel yang digunakan. Sehingga dapat lebih efektif dan efisien dalam mengirim informasi dan tidak terganggu oleh adanya kabel-kabel.

Tabel 3.2. Spesifikasi HC-05

Tegangan Masukan	3-5 VDC
Sensitivitas	-80dBm
Daya Transmit RF	+4dBm
Dimensi	37,3 mm x 15,5 mm
Suhu Maksimum Saat Beroperasi	+75°C
Suhu Minimum Saat Beroperasi	-20°
Frekuensi	2,4GHz



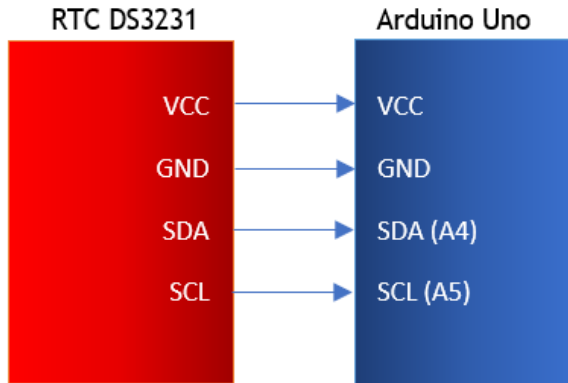
Gambar 3.7 Antarmuka HC-05 dengan Arduino Uno

3.2.5. RTC (Real Time Clock)

RTC adalah sebuah perangkat elektronik berupa 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 berupa jam.

Tabel 3.3. Spesifikasi RTC

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



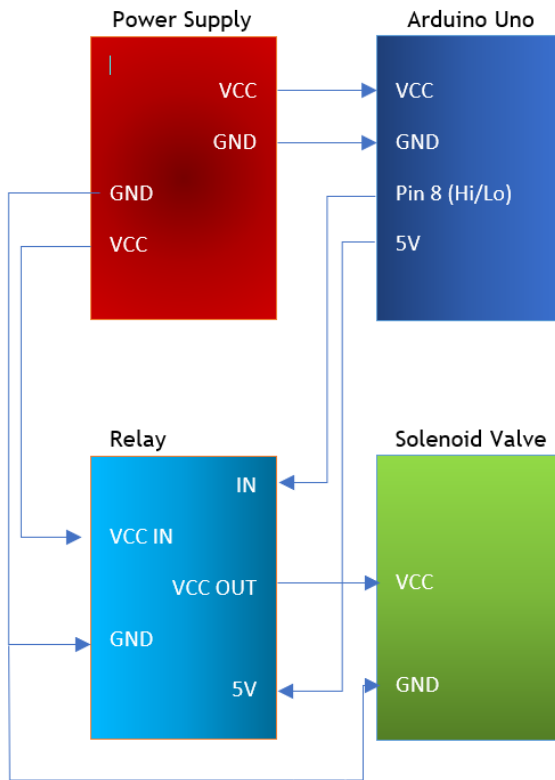
Gambar 3.8 Antarmuka RTC dengan Arduino Uno

3.2.6. Relay dan *Solenoid Valve*

Relay adalah saklar yang dioperasikan secara listrik. Relay menggunakan prinsip elektromagnetik untuk menggerakkan kontak saklar sehingga dengan arus listrik yang kecil dapat menghantarkan listrik yang bertegangan tinggi.

Pada rangkaian ini, relay digunakan sebagai saklar untuk mengatur hidup/matinya *Solenoid Valve* berdasarkan perintah dari mikrokontroler. Jika tanah dalam kondisi kering, maka mikrokontroler akan memberikan masukan ke relay agar relay on, sehingga membuka *Solenoid Valve*, dan mengalirkan air ke tanah.

Solenoid Valve merupakan *valve* yang dikendalikan dengan arus listrik baik AC maupun DC melalui kumparan / selenoida. Jika terhubung dengan listrik, maka *valve* akan terbuka dan mengalirkan air, sedangkan jika tidak dialiri listrik, *valve* akan tertutup dan menahan aliran air. Pembukaan solenoid valve tergantung dari hasil pembacaan sensor, jika hasil pembacaan sensor menunjukkan bahwa tanah kering, maka solenoid valve akan terbuka selama sekian detik, begitu pula jika terbaca Agak basah, maka solenoid valve akan membuka selama sekian detik, namun lebih sebentar dari bukaan valve saat tanah kering. Saat tanah basah, maka relay off, dan solenoid valve tidak membuka sama sekali.



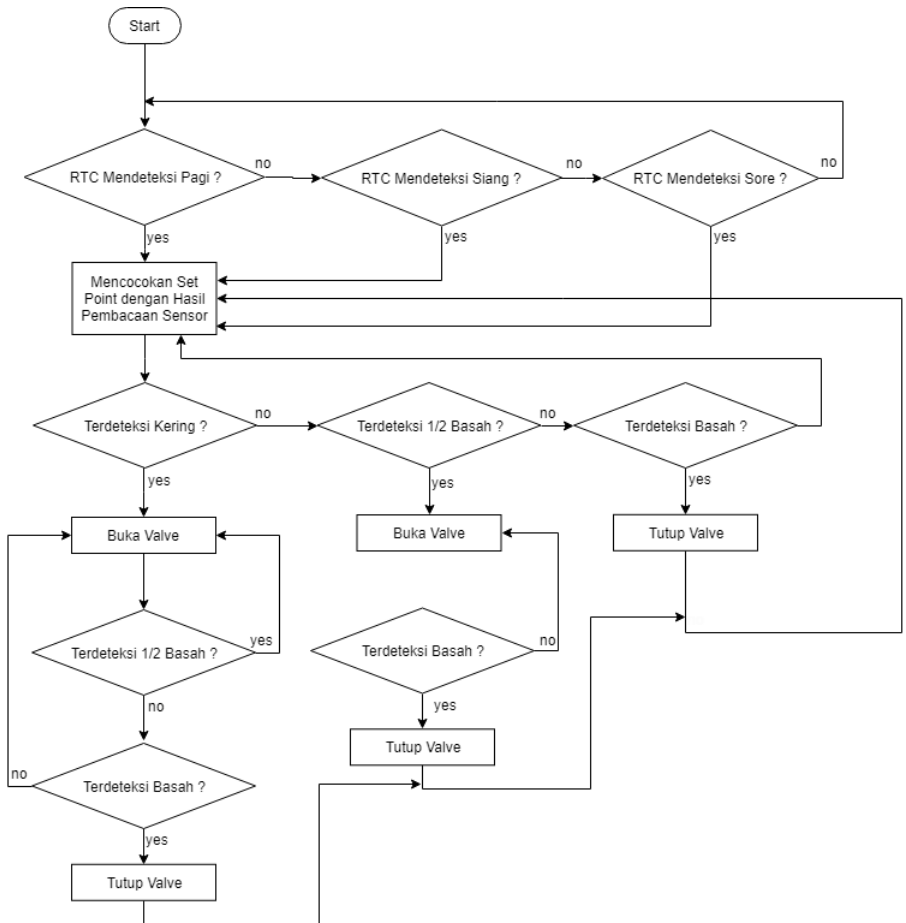
Gambar 3.9 Antarmuka Relay dan *Solenoid Valve* dengan Arduino Uno

3.2.7. Arduino Uno

Arduino Uno berfungsi sebagai mikrokontroller yang membaca data keluaran dari sensor, kemudian memprosesnya. Arduino Uno juga digunakan untuk mengontrol aktuator, dimana aktuatornya berupa *Solenoid Valve*.

3.3. Perancangan Perangkat Lunak

Perancangan perangkat lunak pada tugas akhir ini meliputi pembacaan sensor AMG8833, kontrol *Fuzzy Logic* pada sistem hidup / mati *valve*, dan sistem keseluruhan.



Gambar 3.10 Flowchart Sistem Keseluruhan

3.3.1. Pembacaan Sensor AMG8833

Fungsi dari sensor pada tugas akhir ini sebenarnya cukup sederhana, yaitu sebagai pendeteksi suhu permukaan tanah. Sensor AMG8833 memiliki output berupa nilai suhu permukaan tanah, dibantu dengan aplikasi processing, hasil pembacaan sensor akan ditampilkan ke dalam 64 piksel dan diberikan warna yang berbeda untuk setiap perbedaan suhu

tertentu. Hal ini akan memudahkan untuk menentukan bagian mana yang masih terdeteksi kering, sedang ataupun basah. Hasil dari ke 64 piksel ini juga dirata-ratakan untuk menentukan apakah nantinya akan dilakukan penyiraman atau tidak. Jika penyiraman dilakukan, maka mikrokontroller akan mengaktifkan solenoid valve, sehingga air akan keluar dan mengalir tanah yang kering.

3.3.2. Sistem Kontrol *Fuzzy Logic*

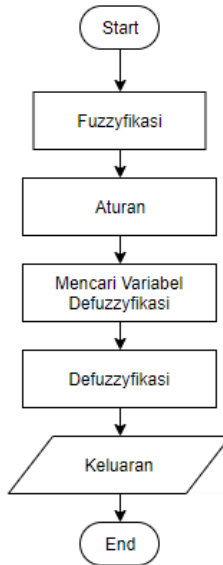
Pada tugas akhir ini, sistem kontrol yang digunakan adalah kontrol *Fuzzy Logic*. Sistem ini digunakan karena konsep logika *fuzzy* mudah dimengerti, fleksibel, dan memiliki toleransi terhadap data-data yang tidak tetap. Pada tugas akhir ini, input yang digunakan berjumlah 2 buah, yaitu sensor AMG8833 dan DS3231. Nilai dari sensor tersebut akan berpengaruh terhadap lama bukaan valve. Langkah pertama dari perancangan sistem *fuzzy logic* ini adalah fuzzyfikasi, dimana fuzzyfikasi merupakan penentuan fungsi keanggotaan. Pada percobaan di dalam ruangan, hanya dibutuhkan 1 set point saat melakukan penyiraman, dikarenakan tidak ada perbedaan suhu antara pagi, siang, dan sore. Sedangkan untuk percobaan di lahan dibutuhkan 3 set point karena suhu permukaan tanah di kondisi yang sama, namun waktu yang berbeda memiliki perbedaan nilai. Sebagai contoh untuk tanah kering di pagi, siang, dan sore masing-masing memiliki suhu yang berbeda. Karena itu, dibuat hanya 1 set point, yaitu set point untuk siang hari, karena respon sensor berbeda-beda untuk setiap waktu yang berbeda, padahal kondisi tanah sama.

Untuk percobaan di dalam ruangan didapatkan set point sebesar 20°C, dimana untuk mendapatkan set point, maka dilakukan kalibrasi dengan sensor kelembapan, sehingga dapat diketahui bagaimana respon sensor saat tanah kering, basah, maupun Agak basah. Untuk percobaan di luar ruangan, didapatkan set pointnya sebesar 28°C.

Input pada sistem fuzzy logic ini adalah error dan Δ error. Dimana error merupakan hasil rata-rata keseluruhan piksel dikurangi dengan set point. Sedangkan Δ error adalah error sekarang dikurangi error sebelumnya.

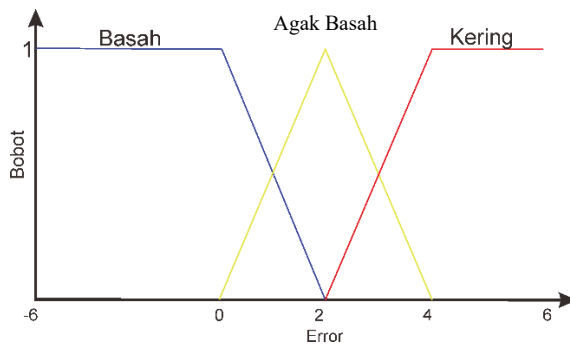
Setelah menentukan input pada sistem fuzzy, maka selanjutnya bisa dilakukan perancangan pada tahap fuzzyfikasi. Tahap fuzzyfikasi adalah tahap menentukan fungsi keanggotaan dari input yang ada. Pada hal ini, akan ditentukan fungsi keanggotaan untuk error dan Δ error. Fungsi keanggotaannya adalah error saat keadaan basah, Agak basah dan kering. Begitu pula dengan Δ error, fungsi keanggotaannya adalah basah, Agak

basah, dan kering.

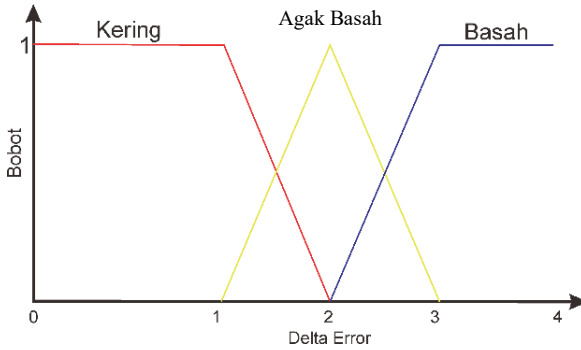


Gambar 3.11 Flowchart Sistem Fuzzy Logic.

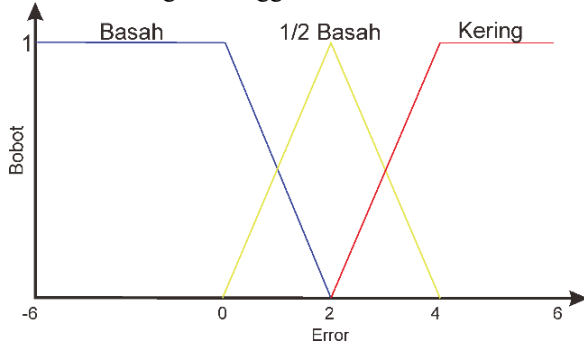
Tahap pertama dari perancangan fuzzy logic adalah fuzzyfikasi, yaitu penentuan fungsi keanggotaan. Fungsi keanggotaan dari sistem fuzzy yang telah dirancang dapat dilihat pada gambar berikut :



Gambar 3.12 Fungsi Keanggotaan error di Dalam Ruang.



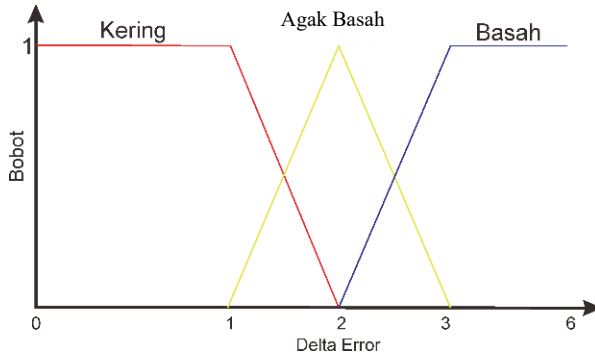
Gambar 3.13 Fungsi Keanggotaan Δ error di Dalam Ruangan



Sistem kontrol di dalam ruangan hanya memiliki 2 jenis dikarenakan percobaan di dalam ruangan hanya menggunakan 1 set point. Maka fungsi keanggotaan di dalam ruangan hanya ada 2, yaitu error dan Δ error. Pada percobaan di lahan, juga hanya menggunakan 2 fungsi keanggotaan, yaitu error dan Δ error. Untuk penyiraman di luar ruangan, dilakukan hanya pada saat siang saja karena nilai pembacaan sensor untuk keadaan yang sama, contoh kering, namun di waktu yang berbeda (pagi, siang, sore) memiliki hasil yang berbeda juga, atau dengan kata lain panas matahari mempengaruhi hasil pembacaan sensor. Maka dari itu hanya akan dilakukan sekali penyiraman, yaitu hanya pada saat siang hari.

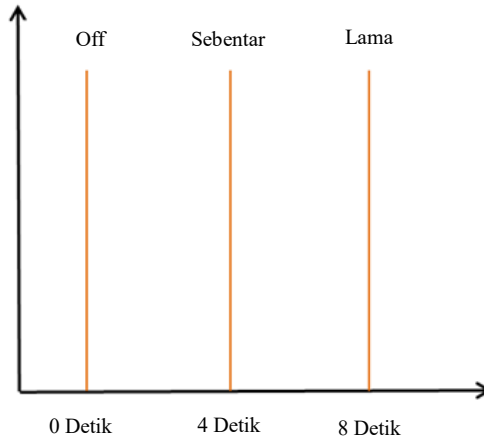
Agak Basah

Gambar 3.14 Fungsi Keanggotaan error Saat Siang Hari



Gambar 3.15 Fungsi Keanggotaan Δ error saat siang hari

Dua gambar di atas adalah fungsi keanggotaan saat siang hari, dengan nilai set point 28.



Gambar 3.16 Fungsi Keanggotaan Output

Gambar di atas adalah gambar fungsi keanggotaan output, dimana, output terdiri dari tiga jenis, yaitu off, dimana valve tidak membuka sama sekali, kemudian sebentar, valve membuka selama 4 detik, dan lama, dimana valve membuka selama 8 detik.

Tabel 3.4. Aturan Dasar (Rule Base)

Δ error \ error	Basah	Agak Basah	Kering
Basah	Off	Sebentar	Sebentar
Agak Basah	Sebentar	Sebentar	Sebentar
Kering	Sebentar	Sebentar	Lama

Dari tabel di atas dapat dilihat bahwa penyiraman dilakukan sebentar jika tanah pada kondisi Agak basah, penyiraman lama dilakukan pada kondisi kering, dan saat kondisi tanah basah, maka penyiraman tidak dilakukan

Setelah menentukan aturan dasar, maka masuk ke tahap selanjutnya yaitu defuzzyfikasi menggunakan aturan Mamdani, dengan metode COA (Center Of Area). Pada metode ini solusi diperoleh dengan cara mengambil titik pusat daerah fuzzy, secara umum dirumuskan pada persamaan 1 untuk variabel kontinyu dan persamaan 2 untuk variabel diskrit.

$$z = \frac{\int z \mu(z) dz}{\int \mu(z) dz} \quad (1)$$

$$z = \frac{\sum_{j=1}^n z_j \mu(z_j)}{\sum_{j=1}^n \mu(z_j)} \quad (2)$$

BAB IV

PENGUJIAN DAN ANALISIS

Bab ini akan membahas tentang pengujian dari sistem yang di rancang pada bab sebelumnya. Bab ini bertujuan untuk mendapatkan data analisa pada masing-masing pengujian. Pengujian dilakukan per bagian komponen alat hingga sistem keseluruhan meliputi pengujian sensor, sistem fuzzy logic, mikrokontroller, serta pengiriman data. Pengujian ini di lakukan di lahan yang berada di depan Departemen Teknik Kimia Institut Teknologi Sepuluh Nopember, dan di Laboratorium Robot Cerdas Gedung B202 Elektro ITS. Gambar dari alat yang berhasil direalisasikan ditunjukkan pada Gambar 4.1.



Gambar 4.1 Gambar Realisasi Alat di Lahan

4.1. Kalibrasi AMG8833 dengan Sensor Kelembapan

Kalibrasi sensor AMG8833 dilakukan dengan menggunakan sensor kelembapan tanah analog, yang akan menampilkan kondisi tanah kering, Agak basah, atau basah jika ditancapkan ke tanah yang ingin dicek tingkat kelembapannya. Hasil dari sensor kelembapan akan dibandingkan dengan nilai dari sensor analog, kemudian dilihat hubungannya, bagaimana hubungan antara kelembapan tanah dengan pembacaan kamera termal.



Gambar 4.2 Sensor Kelembapan Tanah Analog

4.1.1. Kalibrasi di Dalam Ruangan

Kalibrasi dalam ruangan dilakukan agar sensor mendapatkan data yang banyak, dan valid, untuk penentuan set point yang akan digunakan. Nilai dari set point digunakan untuk perancangan fuzzy logic, dimana setpoint yang digunakan tergantung dari seberapa nilai kalibrasi dari sensor analog dengan sensor AMG8833.

Kalibrasi dilakukan dengan menggunakan 10 buah sampel tanah dengan kelembapan yang berbeda-beda, dimana tanah yang pertama tidak diberi air sama sekali. Kemudian tanah yang kedua diberikan satu tutup botol air, botol ketiga diberikan 3 tutup botol air, begitu selanjutnya sampai botol keembilan, sehingga didapatkan sampel tanah dengan kondisi kelembapan yang berbeda-beda. Dari hasil percobaan dibuatlah table dan grafik yang menggambarkan hubungan antara kamera termal dengan kelembapan tanah untuk tanah di dalam ruangan, di mana hasil dari hubungan ini dapat digunakan untuk set point pada perancangan sistem kontrolnya.

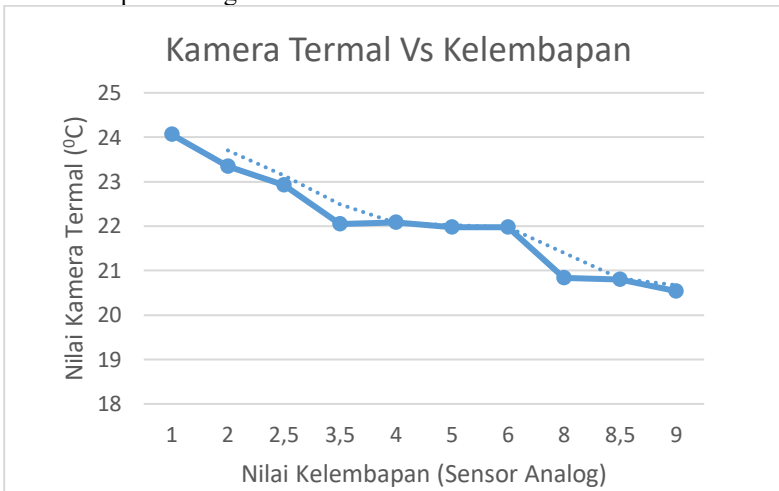


Gambar 4.3 Kalibrasi Tanah di Dalam Ruangan dengan Kelembapan yang Berbeda

Tabel 4.1 Percobaan dengan kelembapan yang berbeda

Siraman (Tutup Botol)	Nilai Sensor Analog (Kelembapan)	Nilai Kamera Termal (°C)
0	1	24,0599
1	2	23,35
2	2,5	22,925
3	3,5	22,05
4	4	22,08333
5	5	21,97292
6	6	21,3356
7	8	20,8375
8	8,5	20,80417
9	9	20,53827

Tabel di atas menunjukkan hubungan antara banyaknya siraman dengan kelembapan dan suhu hasil pembacaan kamera termal. Dari table di atas dibuatlah grafik yang dapat menunjukkan hubungan antara kamera termal dengan kelembapan tanah, dimana sensor yang digunakan adalah sensor kelembapan analog.



Gambar 4.4 Hubungan antara kelembapan dengan kamera termal

Dari grafik di atas dapat dikatakan bahwa semakin lembab tanah di

dalam ruangan, maka suhu yang terdeteksi pada kamera termal semakin kecil.

4.1.2. Kalibrasi di Lahan

Kalibrasi juga dilakukan di lahan, karena memang alat ini juga realisasinya adalah di lahan terbuka, sehingga dibutuhkan data saat alat ini digunakan di lahan terbuka, dan langung terpapar sinar matahari.



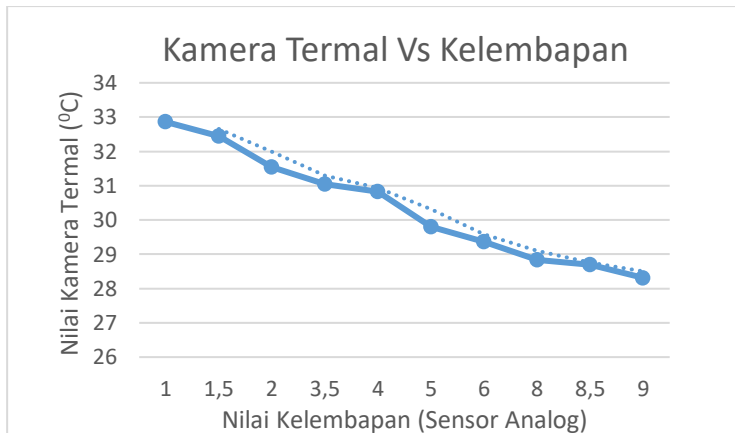
Gambar 4.5 Kalibrasi Tanah di Lahan dengan kelembapan yang berbeda.

Pada kalibrasi di lahan, juga dilakukan dengan menggunakan sampel tanah yang sama, namun diberikan air sehingga tanah yang sedang dipantau memiliki kelembapan yang berbeda-beda.

Tabel 4.2 Percobaan dengan kelembapan yang berbeda

Siraman (Tutup Botol)	Nilai Sensor Analog (Kelembapan)	Nilai Kamera Termal (°C)
0	1	32,85938
2	1,5	32,44922
4	2	31,539
6	3,5	31,043
8	4	30,8333
10	5	29,8052
12	6	29,363
14	8	28,83202
16	8,5	28,69141
18	9	28,3125

Tabel di atas menunjukkan perbedaan banyaknya siraman di antara percobaan di dalam ruangan dan percobaan di lahan. Dari tabel di atas dibuatlah grafik hubungan antara kelembapan tanah dengan kamera termal.



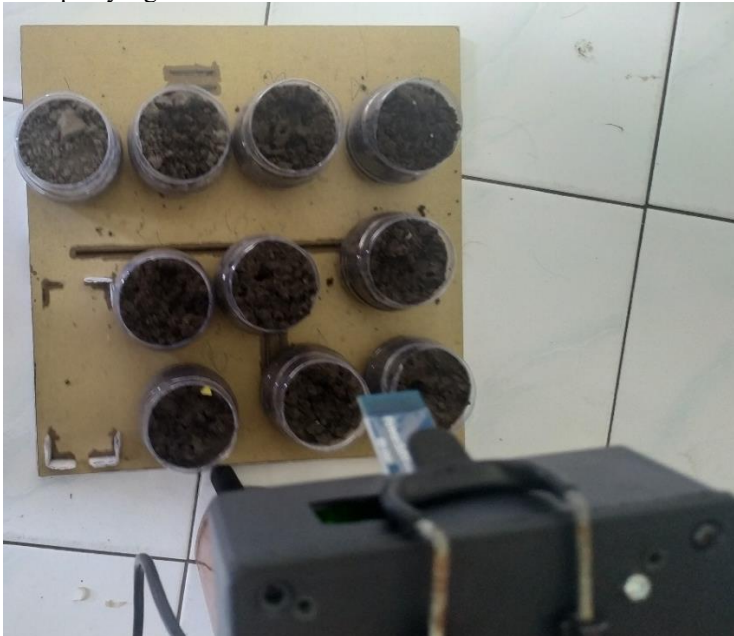
Gambar 4.6 Hubungan antara kelembapan dengan kamera termal

4.2. Pengujian Sensor AMG8833

Sebagai tahap awal pada sistem, harus dipastikan perbedaan hasil pembacaan sensor untuk setiap kondisi yang berbeda, seperti tanah kering, tanah Agak basah, dan tanah basah. Setiap tanah akan dikondisikan dan akan dilihat hasil pembacaan sensornya. Pengujian dilakukan pada tiga kondisi waktu, yaitu pagi, siang, dan sore. Jadi pengujian ini akan melihat pembacaan sensor saat pagi, siang, dan sore hari dengan kondisi tanah kering, Agak basah, dan basah. Data ini dibutuhkan agar nantinya bisa menentukan *set point* pada sistem kontrol *valve*.

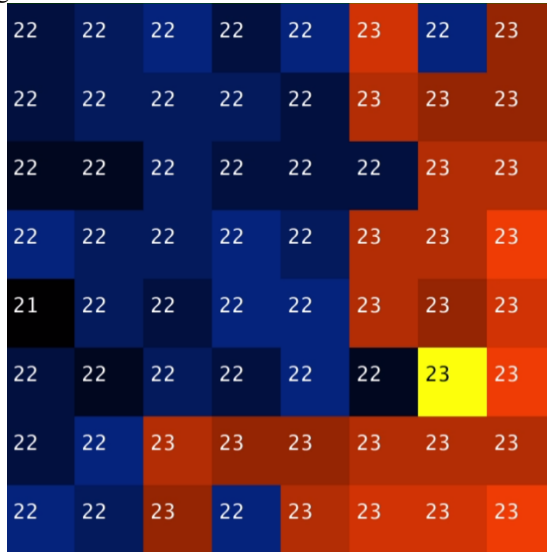
4.2.1. Pengujian di Dalam Ruangan

Pengujian di dalam ruangan dilakukan untuk mengetahui respon sensor AMG8833 terhadap sampel tanah yang sudah memiliki tingkat kelembapan tertentu. Pengujian dilakukan dengan menempatkan sampel tanah pada kamera termal, kemudian hasilnya akan dilihat pada processing, dan dilihat bagaimana responnya terhadap tanah dengan kelembapan yang berbeda-beda.



Gambar 4.7 Sampel Tanah dengan Kelembapan yang berbeda

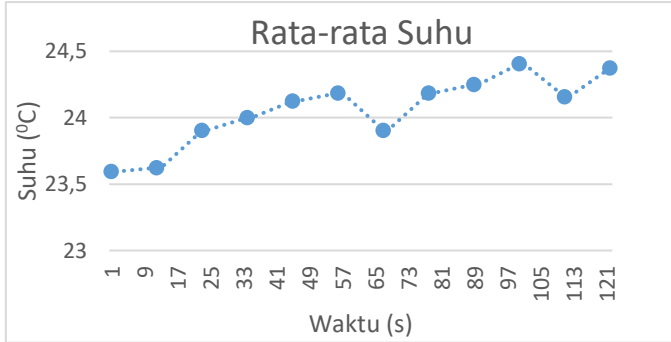
Hasil dari pembacaan sensor ditampilkan menggunakan aplikasi processing, dimana diberikan pewarnaan sesuai dengan hasil rata-rata yang didapatkan oleh sensor. Hasil pembacaan didalam ruangan dapat dilihat pada gambar 4.8



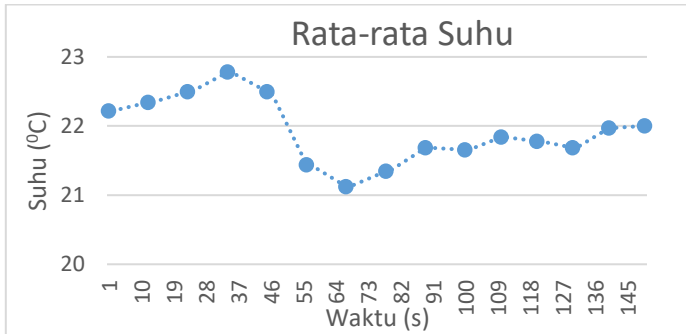
Gambar 4.8 Hasil Pembacaan Sensor di Dalam Ruangan

Warna merah menunjukkan bahwa suhu pada titik tersebut lebih tinggi daripada suhu rata-rata. Sedangkan warna kuning menunjukkan titik dimana suhu maksimum berada. Dari hasil pembacaan sensor AMG8833, dapat dilihat bahwa bagian sebelah kanan terdeteksi lebih kering daripada bagian yang sebelah kiri, dapat dibandingkan juga dengan gambar 4.2, dimana tanah yang kering berada di bagian paling kiri, maka pembacaan sensor cukup menunjukkan bagian yang paling kiri adalah bagian yang kering.

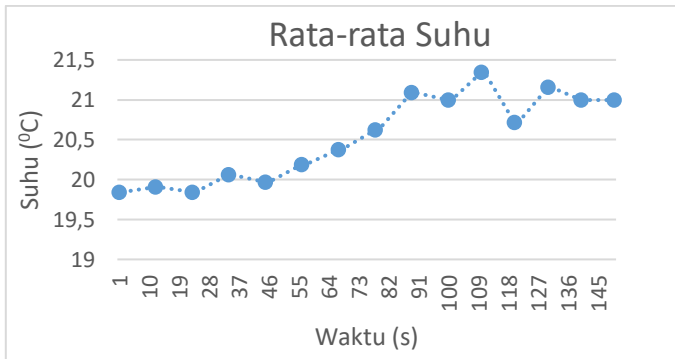
Dari hasil percobaan juga didapatkan grafik dari tanah yang kering, tanah agak basah, dan tanah basah. Dari garfik ini dapat dilihat bahwa untuk waktu yang sama, dan ruangan yang sama, maka hasil pembacaan sensor juga sama, namun jika berpindah tempat, maka kemungkinan pembacaan sensor juga berbeda.



Gambar 4.9 Grafik Rata-rata Suhu Untuk Tanah Kering.

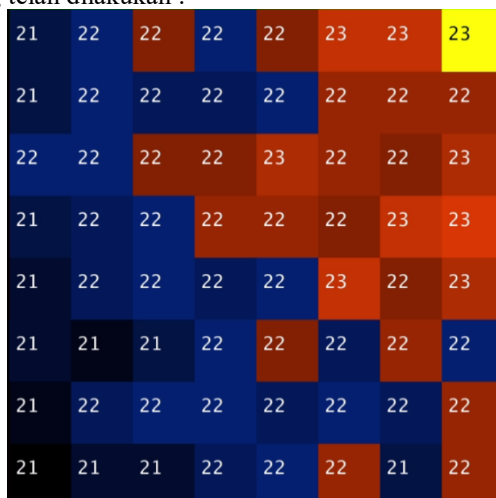


Gambar 4.10 Grafik Rata-rata Suhu Untuk Tanah Agak Basah.

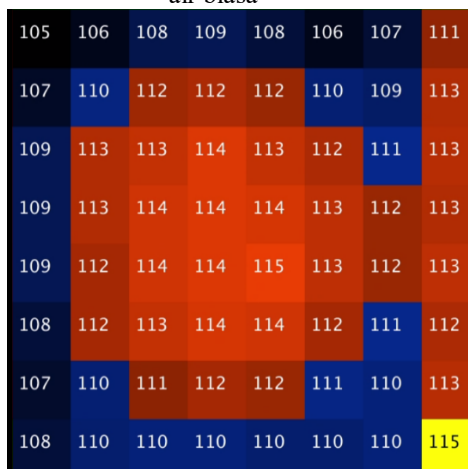


Gambar 4.11 Grafik Rata-rata Suhu Untuk Tanah Basah.

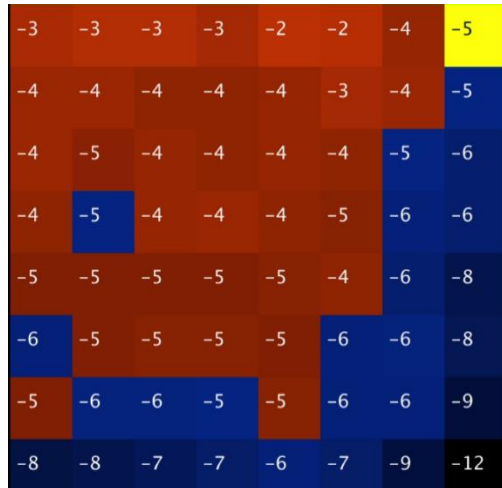
Pengujian di dalam ruangan juga dilakukan dengan menggunakan air panas dan air dingin, untuk melihat bagaimana respon sensor terhadap masing-masing air yang disiramkan ke tanah. Berikut merupakan hasil pengujian yang telah dilakukan :



Gambar 4.12 Hasil Pembacaan Sensor untuk tanah basah dengan air biasa



Gambar 4.13 Hasil Pembacaan Snsor untuk tanah basah dengan air panas

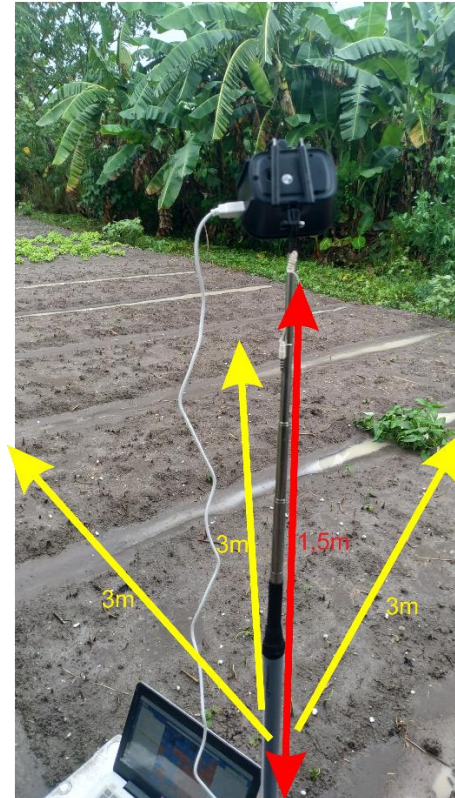


Gambar 4.14 Hasil Pembacaan Sensor untuk tanah basah dengan air dingin

Dari hasil pengujian menggunakan 3 jenis air yang berbeda, yaitu air dengan suhu normal, air panas, dan air dingin dapat dilihat bahwa air juga dapat mempengaruhi hasil pembacaan sensor. Hal ini dapat mempengaruhi seberapa tepat alat mendeteksi tanah yang kering, basah, maupun agak basah. Dari gambar 4.7, gambar 4.8, dan gambar 4.9 dapat dilihat rata-rata untuk tanah basah menggunakan air biasa adalah sekitar 23°C , untuk tanah basah dengan air panas suhu rata-rata tanah yang terbaca mencapai sekitar 113°C , sedangkan untuk tanah basah dengan air dingin suhu rata-rata tanah yang terbaca mencapai sekitar -3°C , dari hasil ini dapat dilihat kelemahan menggunakan kamera termal pada alat ini. Dimana kamera termal mendeteksi suhu air, meskipun tanah tersebut basah, namun jika air yang digunakan adalah air panas, bisa saja alat tidak mendeteksi tanah yang basah, namun tanah tetap terdeteksi kering, akibatnya terjadi kesalahan pendeteksian tanah kering, basah, ataupun agak basah. Namun pada preteknya, air tidak akan sepanas air mendidih, sejauh penulis mengambil data, air yang digunakan pasti akan memberikan pengaruh pada tanah, yaitu membuat suhu tanah lebih rendah dari sebelumnya.

4.2.2. Pengujian di Lahan

Peletakan kamera pada pengujian di lahan awalnya dilakukan secara coba-coba untuk menentukan jarak terbaik, agar kamera dapat membaca suhu permukaan tanah dengan baik. Dari percobaan tersebut didapat bahwa kamera diletakkan dengan ketinggian sekitar 1,5 meter, dimana kamera dapat menangkap suhu permukaan tanah dengan baik pada jarak 3 meter.



Gambar 4.15 Posisi dan jarak pada pengujian AMG8833

Pengujian di lahan dilakukan 3 kali, yaitu saat pagi hari, saat siang hari, dan saat malam hari. Pengujian sensor dilakukan untuk mendapatkan data pembacaan sensor, sehingga diketahui bagaimana respon sensor saat kondisi kering di pagi hari, kondisi kering di siang hari, dan kondisi kering

di sore hari. Begitu pula dengan kondisi Agak basah, dan kondisi basah. Hasil dari pengujian ini akan menentukan model dari kontrol yang akan digunakan. Pengujian di lahan intinya bertujuan untuk mengetahui respon sensor di setiap keadaan, dan di setiap waktu yang berbeda.

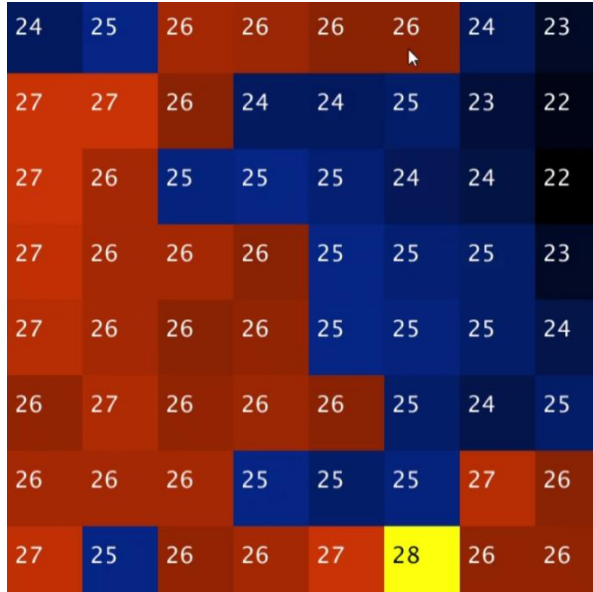
Pengujian di lahan sangat perlu dilakukan karena penggunaan alat ini tujuannya adalah lahan terbuka, dimana lahan yang digunakan juga terpapar sinar matahari langsung. Dari pengujian ini akan dilihat bagaimana hasil pembacaan sensor untuk waktu yang berbeda, dengan kondisi kelembapan yang berbeda-beda pula.

a. Pengujian Saat Pagi Hari

Hasil dari pengujian saat pagi hari dapat dilihat pada gambar di bawah.



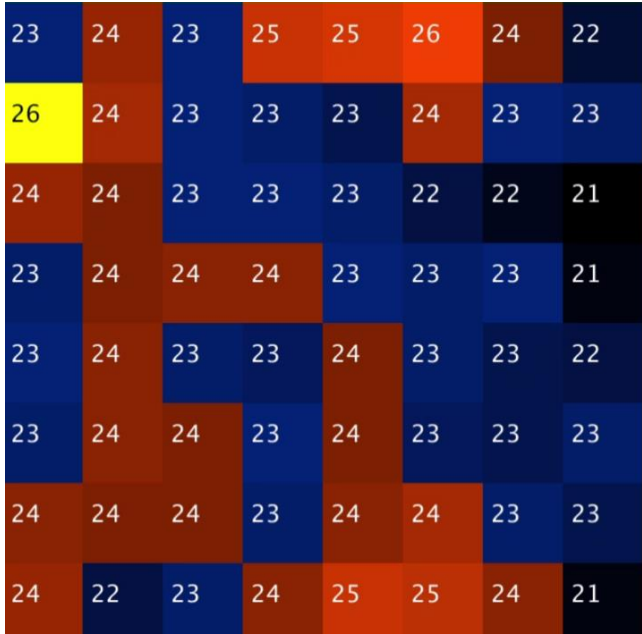
Gambar 4.16 Keadaan Tanah saat Kering



Gambar 4.17 Hasil Pembacaan Sensor saat Tanah Kering



Gambar 4.18 Keadaan Tanag Saat Agak Basah



Gambar 4.19 Hasil Pembacaan Sensor Saat Tanah Agak Basah

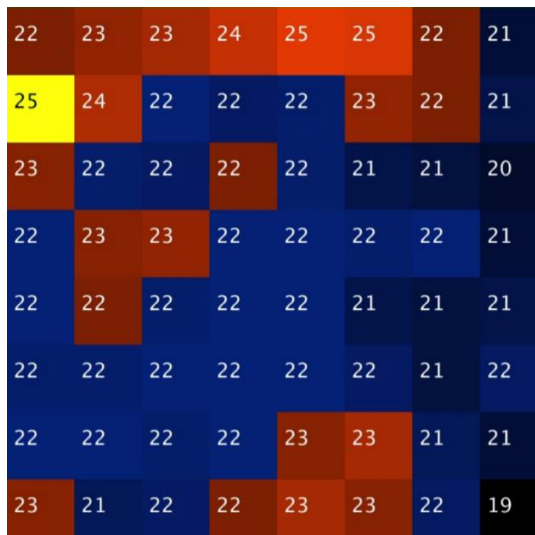


Gambar 4.20 Keadaan Tanah Saat Basah

Pengujian di luar ruangan juga bertujuan untuk melihat respon sensor terhadap penyiraman, dapat dilihat pada gambar di atas, tanah di siram saat pagi hari namun tidak semua bagian tanah dibasahi dengan air. Respon sensor terhadap perbedaan pemberian air dapat dilihat pada gambar 4.21.

Gambar 4.21 menunjukkan bahwa bagian yang berwarna merah adalah bagian yang belum disiram, sedangkan bagian yang berwarna biru adalah bagian yang sudah disiram dengan air. Terdapat perbedaan pembacaan sensor pada bagian yang telah disiram dan yang belum disiram. Bagian yang disiram berwarna lebih biru.

Hal ini menunjukkan bahwa sensor dapat mendeteksi bagian yang belum disiram, dengan cara mendeteksi suhu permukaan mana yang belum disiram dengan air. Percobaan ini juga dilakukan pada saat siang hari dan sore hari, hasilnya juga dapat dilihat pada gambar di bawah. Pengujian sensor AMG8833 bertujuan untuk melakukan pengecekan pada respon sensor pada keadaan yang sama, untuk intensitas cahaya matahari yang berbeda. Didapatkan bahwa semakin panas matahari, maka nilai pembacaan sensor juga semakin besar. Hal inilah yang menyebabkan perbedaan pada pembacaan sensor pada keadaan yang sama, namun di waktu yang berbeda, yaitu panas matahari.



Gambar 4.21 Hasil Pembacaan Sensor saat Tanah Basah

b. Pangujian Saat Siang Hari

Hasil dari pengujian saat Siang hari dapat dilihat pada gambar di bawah.



Gambar 4.22 Keadaan Tanah saat Kering

29	29	30	30	28	28	28	28
31	33	35	35	34	32	29	27
33	33	32	31	30	32	32	29
29	29	29	30	29	31	31	30
28	30	31	28	28	30	30	29
28	28	28	28	28	29	30	29
27	27	28	29	28	28	29	29
26	27	28	29	29	27	29	28

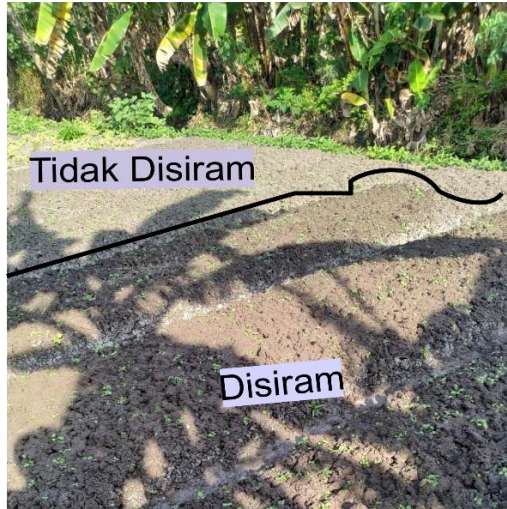
Gambar 4.23 Hasil Pembacaan Sensor saat Tanah Kering



Gambar 4.24 Keadaan Tanah Saat Agak Basah

28	27	28	28	27	27	26	26
28	31	33	32	32	29	27	26
30	26	27	27	26	27	28	27
25	24	25	26	25	25	27	27
25	25	25	25	25	25	26	26
25	25	25	26	25	25	26	25
25	25	25	27	26	25	26	25
24	24	25	27	26	25	25	24

Gambar 4.25 Hasil Pembacaan Sensor Saat Tanah Agak Basah



Gambar 4.26 Keadaan Tanah Saat Basah

Pengujian pada siang hari juga menunjukkan hasil yang sama dengan pengujian pada pagi hari, namun suhu yang didapatkan lebih tinggi daripada suhu di pagi hari.

Dapat juga dilihat ada bagian yang tidak disiram, dan terlihat pada pembacaan sensor bahwa bagian yang tidak disiram berwarna merah, sedangkan bagian yang sudah terkena siraman air berwarna biru. Hasil pembacaan sensor menunjukkan perbedaan pada bagian tanah yang sudah disiram dan yang belum terkena siraman air.

Dari hasil percobaan juga dapat dilihat perbedaan hasil pembacaan sensor untuk siang dan pagi hari, padahal kondisi tanah sama-sama basah. Dapat dilihat pada siang hari, saat basah suhu tanah mencapai 24°C sedangkan saat pagi hari saat basah suhu hanya mencapai 22°C. Dari hasil ini dapat dilihat bahwa adanya perbedaan pembacaan untuk pagi, dan siang hari. Hal ini wajar dikarenakan panas matahari memang berbeda untuk masing-masing keadaan baik itu pagi maupun siang hari. Dari hasil pembacaan sensor juga dapat dilihat suhu kering tanah pada siang hari mencapai 35°C, jika dibandingkan dengan suhu kering pagi hari, sangat jauh perbedaannya.

27	27	27	27	26	27	26	27
28	30	31	31	31	29	27	26
29	26	26	26	25	26	28	26
25	24	24	25	24	25	26	27
25	25	24	24	24	25	26	25
25	25	24	26	25	25	26	25
25	25	25	26	27	25	26	25
24	24	25	26	26	25	25	25

Gambar 4.27 Hasil Pembacaan Sensor saat Tanah Basah

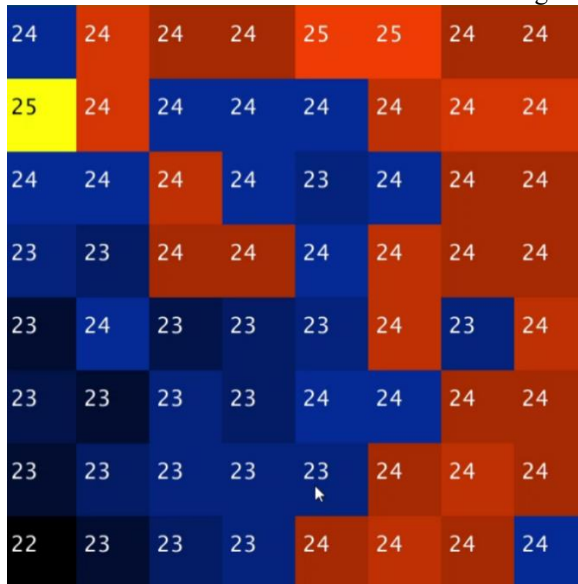
Dari hasil pembacaan sensor dapat dilihat pada saat tanah basah perbedaan suhunya dapat dilihat cukup jauh. Terlihat pada gambar, tanah yang basah berada pada suhu sekitar 24°C, sedangkan untuk tanah yang kering terlihat jauh perbedaannya dengan tanah yang basah, dimana tanah yang kering berada pada kisaran suhu 31°C. Maka dari itu, sangat memungkinkan untuk memilih siang hari menjadi waktu penyiraman, karena range dari tanah yang kering ke tanah yang basah cukup besar, jika dibandingkan dengan range kering-basah di pagi hari, perbedaan pada pagi hari sangat kecil, sehingga untuk membedakan tanah kering, dan tanah basah cukup sulit. Maka dari itu, kemungkinan penyiraman akan dilakukan pada siang hari, karena menjadi lebih mudah untuk membedakan kondisi tanah kering, basah, maupun agak basah. Namun melihat juga bagaimana hasil pembacaan di sore hari tergantung bagaimana range antara basah-kering apakah sebesar range di siang hari, jika memang rangnya lebih kecil dari range di siang hari, maka kemungkinan penyiraman akan dilakukan hanya pada siang hari.

c. Pengujian Saat Sore Hari

Hasil dari pengujian saat Sore hari dapat dilihat pada gambar di bawah.



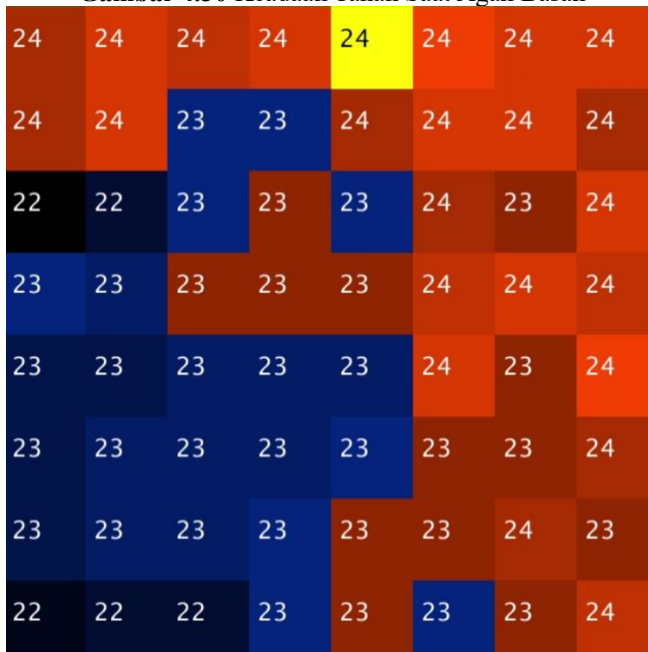
Gambar 4.28 Keadaan Tanah saat Kering



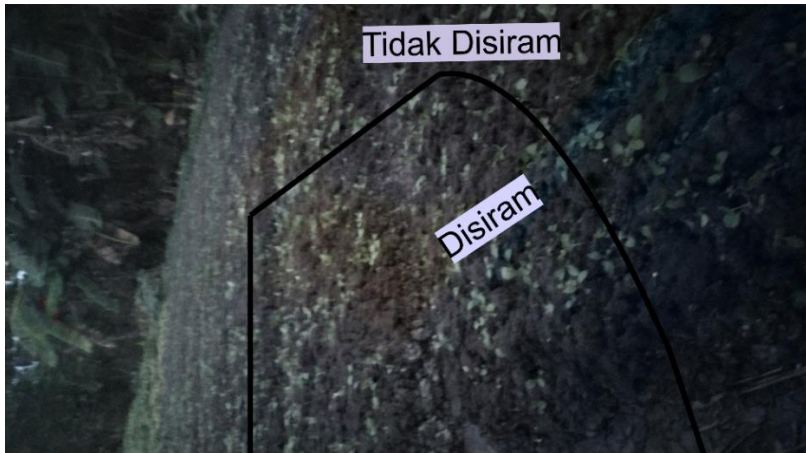
Gambar 4.29 Hasil Pembacaan Sensor saat Tanah Kering



Gambar 4.30 Keadaan Tanah Saat Agak Basah



Gambar 4.31 Hasil Pembacaan Sensor Saat Tanah Agak Basah

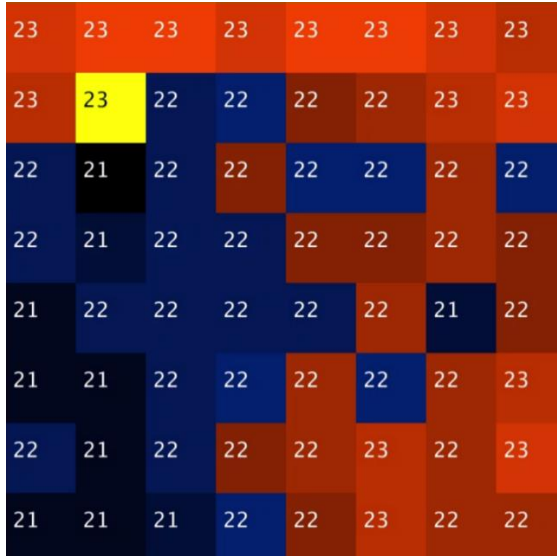


Gambar 4.32 Keadaan Tanah Saat Basah

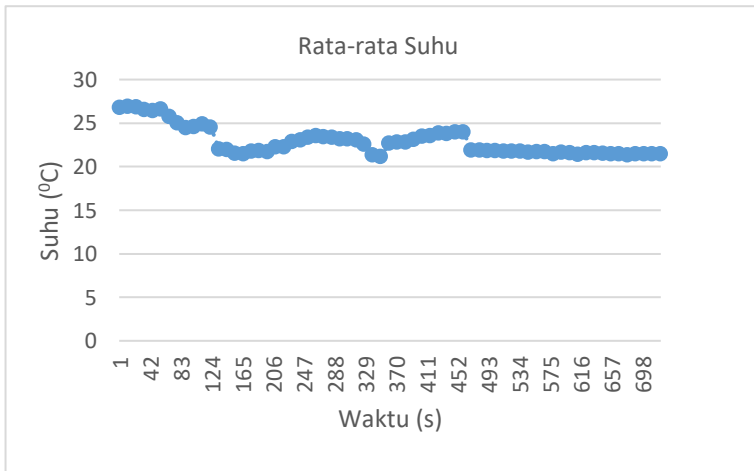
Pada percobaan sore dapat dilihat bahwa pembacaan sensor dapat dipengaruhi oleh matahari. Semakin panas matahari, maka nilai pembacaan sensor menjadi sangat besar, sedangkan semakin dingin matahari, maka nilai pembacaan sensor juga semakin kecil. Hal ini mengakibatkan pembuatan set point untuk sistem fuzzy logic harus menggunakan satu set point, yaitu pada siang hari saja, artinya penyiraman hanya akan dilakukan pada siang hari saja.

Dapat juga dilihat pada pembacaan sensor ada bagian tanah yang disiram dan tidak disiram. Hasil pembacaan sensor menunjukkan dengan jelas bahwa bagian berwarna merah berarti belum disiram, sedangkan bagian berwarna biru telah disiram. Pada gambar juga dapat dilihat pembacaan sensor untuk sore hari adalah yang paling kecil diantara pembacaan sensor saat pagi hari dan pembacaan sensor saat siang hari. Dapat dibandingkan antara gambar 4.23 dengan gambar 4.12 dan gambar 4.18, dapat dilihat bahwa pembacaan sensor pada sore hari adalah yang paling kecil. Mencapai rata-rata 22°C , sedangkan untuk pagi hari rata-ratanya masih sekitar 23°C , dan untuk siang hari, rata-ratanya mencapai 25°C .

Setelah percobaan, didapatkan grafik perubahan suhu tanah dari kerning, agak basah, sampai basah namun di waktu berbeda, grafik dapat dilihat setelah gambar di bawah, dimana grafik terdiri dari masing-masing pagi, siang dan sore.

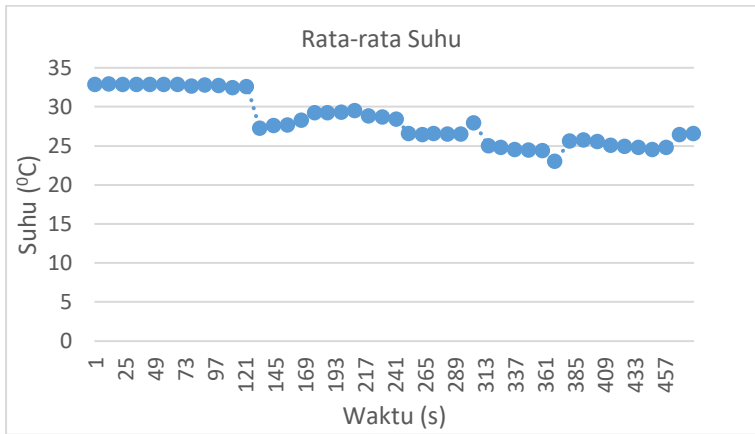


Gambar 4.33 Hasil Pembacaan Sensor saat Tanah Basah
Setelah pengujian sensor dilakukan, maka didapatkan hasil sebagai berikut :

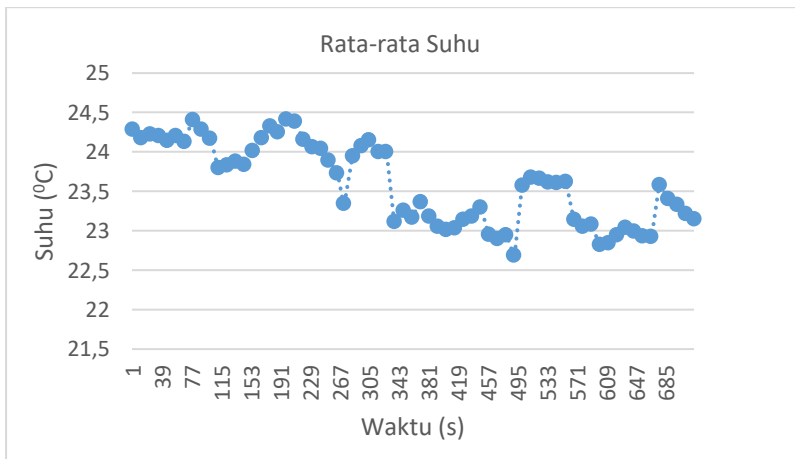


Gambar 4.34 Grafik Hasil Pembacaan Sensor dari kondisi kering ke

kondisi basah di pagi hari



Gambar 4.35 Grafik Hasil Pembacaan Sensor dari kondisi kering ke kondisi basah di siang hari

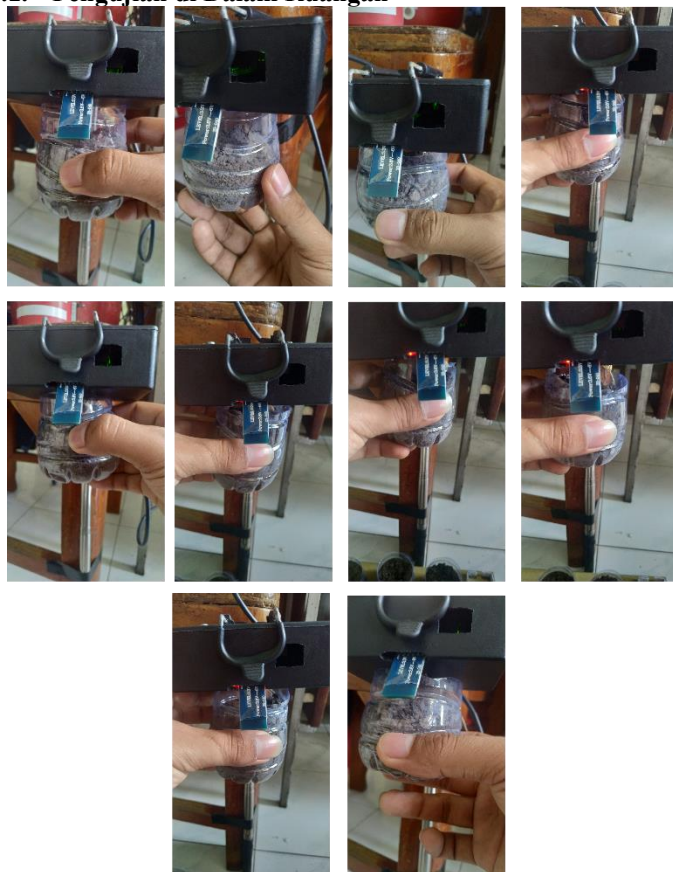


Gambar 4.36 Grafik Hasil pembacaan sensor dari kondisi kering ke kondisi basah di sore hari

4.3. Pengujian Sistem *Fuzzy Logic*

Pengontrolan valve dilakukan dengan menggunakan sistem kontrol fuzzy logic, dengan menggunakan set point yang berbeda untuk masing-masing pagi, siang, maupun sore hari. Pengujian fuzzy logic meliputi pengujian lama waktu bukaan valve terhadap suhu hasil bacaan sensor. Hasil pengujian dapat dilihat pada tabel di bawah.

4.3.1. Pengujian di Dalam Ruangan

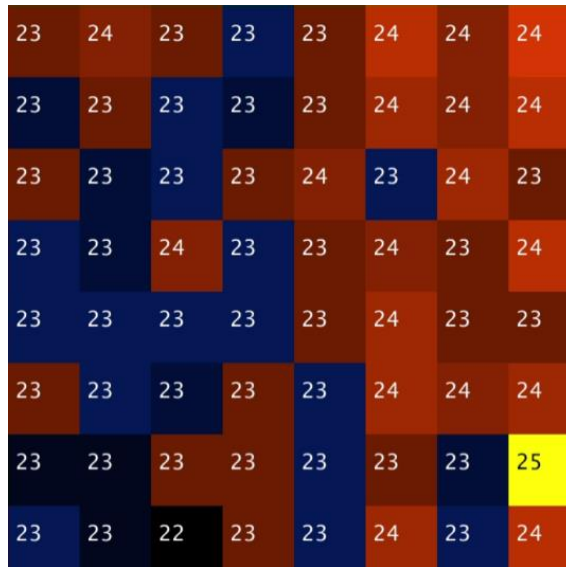


Gambar 4.37 Pengujian di Dalam Ruangan.

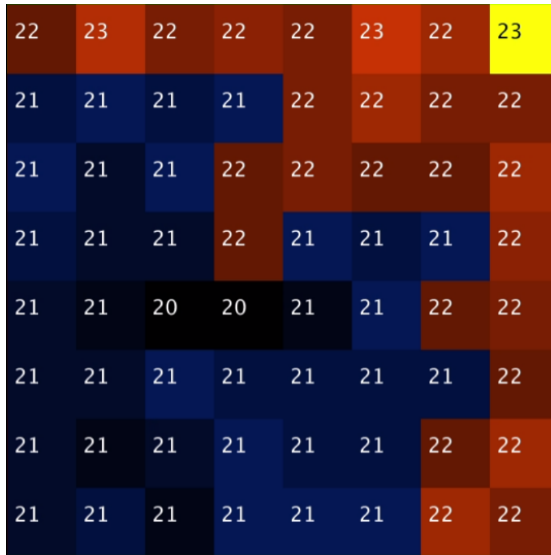
Pengujian di dalam ruangan dapat dilihat pada gambar di atas di mana tanah dengan kelembapan yang berbeda-beda di dekatkan ke sensor. Hasilnya dapat dilihat pada tabel dibawah ini dimana dilakukan percobaan dengan menggunakan tanah dengan kelembapan yang berbeda-beda. Dari tabel dapat dilihat, untuk tanah basah, maka suhu tanah yang terdeteksi adalah sekitar 19°C, dimana tidak akan dilakukan penyiraman, atau dengan kata lain valve tidak terbuka. Selanjutnya untuk kondisi tanah agak basah, yaitu berada pada kisaran 21-22°C, valve pun akan terbuka namun untuk waktu yang berada di tengah-tengah dari waktu maksimum terbukanya valve. Untuk suhu 24°C maka valve akan terbuka dengan waktu maksimum yaitu 8 detik. Ini merupakan pengujian di dalam ruangan, di mana tanah tidak terkena panas langsung dari matahari.

Tabel 4.3 Hasil Pengujian Sistem Fuzzy Logic di dalam ruangan

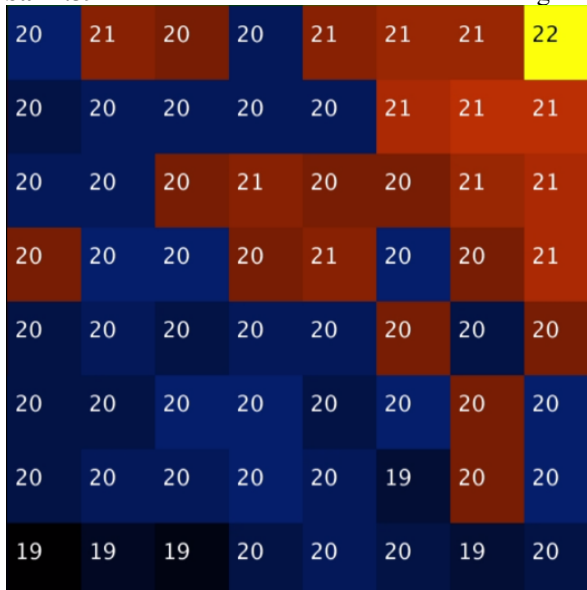
Suhu Rata-rata (°C)	24	23	22	21	20	19
Lama Buka Valve (detik)	8	7	5	3	2	0



Gambar 4.38 Hasil Pembacaan Sensor Untuk Tanah Kering.



Gambar 4.39 Hasil Pembacaan Sensor Untuk Tanah Agak Basah.



Gambar 4.40 Hasil Pembacaan Sensor Untuk Tanah Basah.

4.3.2. Pengujian di Lahan

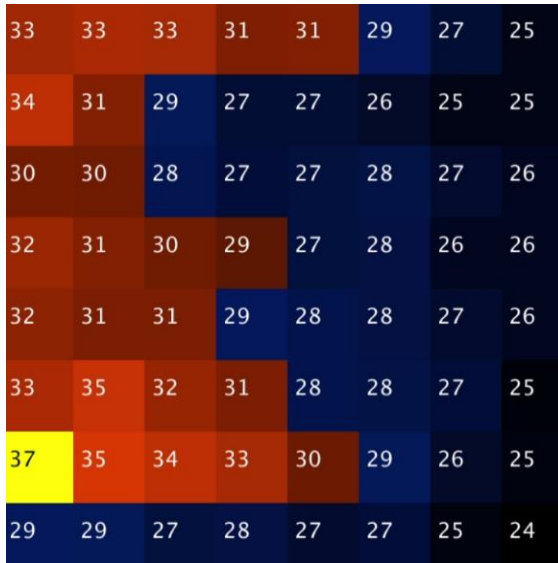
Pengujian sistem fuzzy logic di lahan dilakukan dengan menggunakan satu waktu saja, agar tidak terganggu oleh perbedaan suhu yang diakibatkan oleh panas matahari. Setelah melihat hasil pada masing-masing, baik itu pagi, siang, maupun sore, maka ditentukan bahwa penyiraman hanya akan dilakukan pada saat siang hari saja. Pengujian sistem fuzzy logic pada siang hari di lahan dapat dilihat pada tabel di bawah ini. Pada suhu rata-rata 31°C, tanah terdeteksi basah, sehingga penyiraman tidak dilakukan, atau dengan kata lain valve tidak terbuka. Pada rentang 32-34°C, tanah terdeteksi agak basah, sehingga penyiraman hanya dilakukan sedikit saja, atau dapat dikatakan valve hanya terbuka sebentar saja. Pada suhu 35-37°C tanah terdeteksi kering, sehingga valve juga terbuka cukup lama, hingga waktu maksimal yaitu 8 detik.

Tabel 4.4 Hasil Pengujian Sistem Fuzzy Logic pada Siang Hari

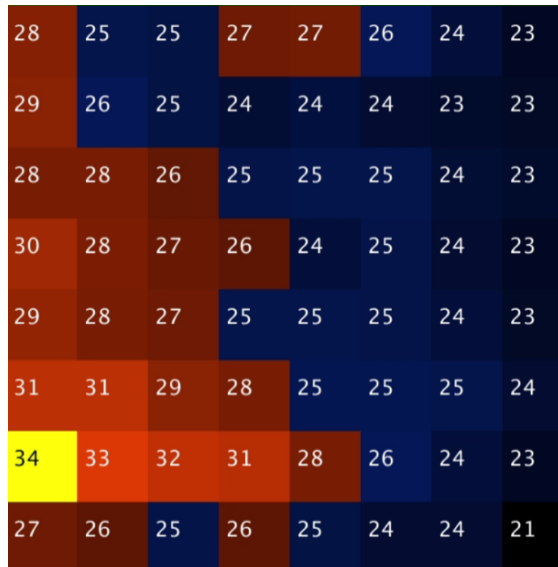
Suhu Rata-rata (°C)	37	36	35	34	33	32	31
Lama Bukaannya Valve (detik)	8	7	6	5	4	2	0



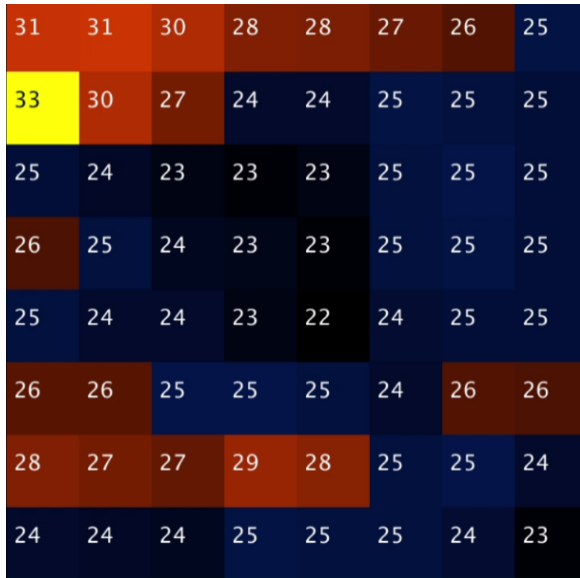
Gambar 4.41 Valve Terbuka



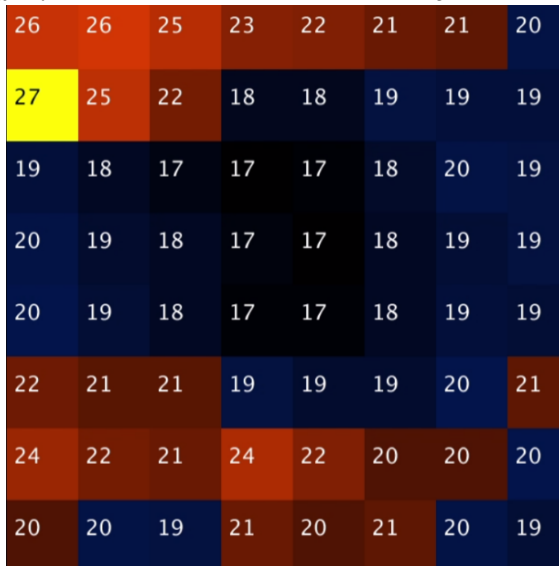
Gambar 4.42 Hasil Pembacaan Sensor Sebelum disiram



Gambar 4.43 Hasil Pembacaan Sensor Setelah 3 detik Penyiraman



Gambar 4.44 Hasil Pembacaan Sensor Setelah 5 Detik Penyiraman



Gambar 4.45 Hasil Pembacaan Sensor Setelah 8 detik Penyiraman

4.4. Pengujian Keseluruhan Sistem

Pengujian Keseluruhan Sistem dilakukan untuk mengevaluasi sistem secara menyeluruh. Apakah semuanya sudah berjalan dengan baik sesuai dengan yang direncanakan atau bahkan sebaliknya. Pengujian keseluruhan sistem meliputi pengujian terhadap sistem bukaan *valve*, yang menggunakan *fuzzy logic*, yang telah digabungkan dengan DS3231.



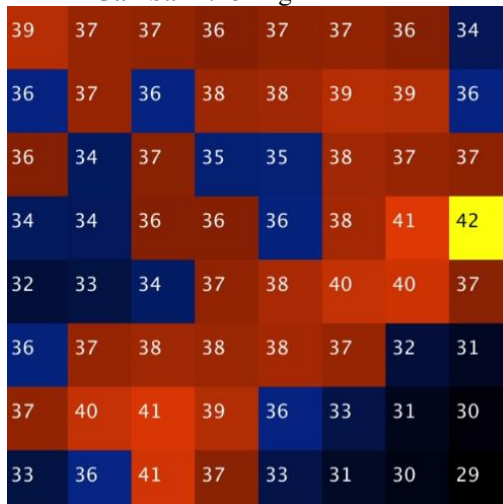
Gambar 4.46 Keseluruhan Sistem



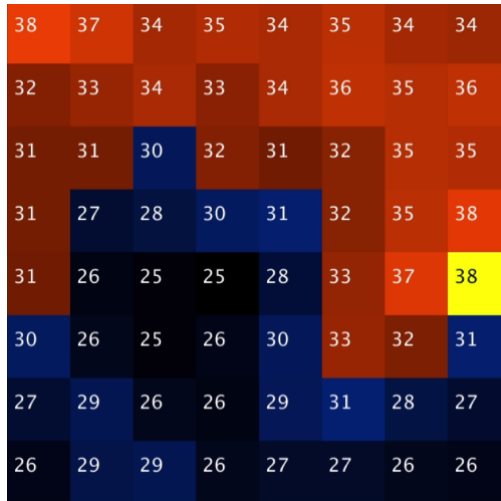
Gambar 4.47 Hasil Pada LCD



Gambar 4.48 Bagian Receiver



Gambar 4.49 Hasil Pembacaan Sensor Saat Tanah Kering



Gambar 4.50 Hasil Pembacaan Sensor Setelah dilakukan penyiraman

Pada pengujian keseluruhan sistem juga dilakukan pengujian tentang seberapa tepat alat mendeteksi tanah kering, agak basah, maupun tanah basah. Juga dilakukan pengujian tentang seberapa baik alat mengontrol kelembapan tanah. Dilakukan sekitar 30 kali pengujian di dalam ruangan, dan di luar ruangan sekitar 10 kali pengujian. Pengujian ini dibandingkan dengan nilai pembacaan dari sensor analog, dimana, sensor analog ini memiliki range nilai untuk kelembapan tertentu. Untuk nilai 0-3 tanah ini merupakan tanah yang kering, untuk nilai 4-7 tanah berada pada kondisi agak basah, untuk nilai 7-10 tanah berada pada kondisi basah. Hasil pengujian dapat dilihat pada table 4.5 dimana akan ditampilkan nilai pembacaan sensor analog, kemudian dibandingkan dengan pembacaan sensor AMG8833.

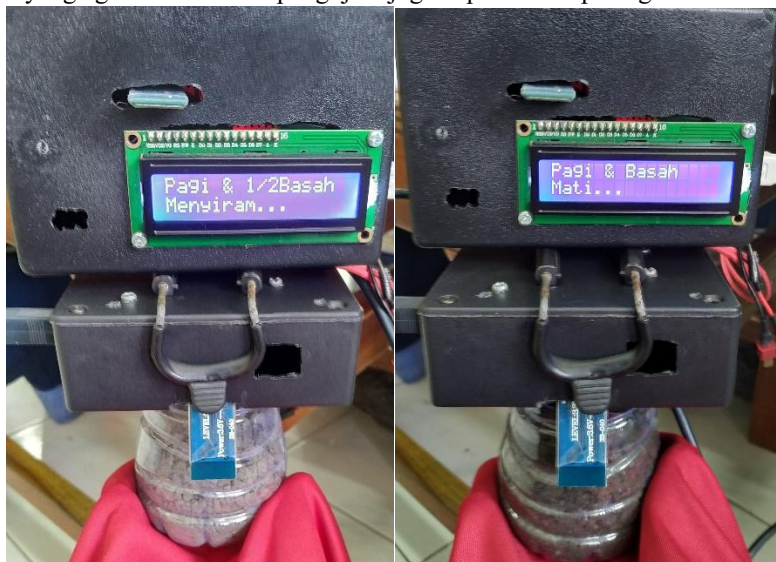
Tabel 4.5 Pengujian sensor dalam mendeteksi kelembapan

Nilai Sensor Analog	Hasil Pembacaan Sensor
1	Kering
2	Kering
3	Kering
4	Kering
5	Agak Basah

6	Kering
7	Agak Basah
8	Agak Basah
9	Basah
10	Basah
1,5	Kering
2,5	Kering
3,5	Kering
4,5	Agak Basah
5,5	Agak Basah
6,5	Agak Basah
7,5	Agak Basah
8,5	Basah
9,5	Basah
4	Kering
2	Kering
5	Agak Basah
7	Basah
1	Kering
8	Basah
6	Agak Basah
9	Basah
10	Basah
3	Agak Basah
0	Kering

Dari tabel di atas dapat dilihat bahwa terdapat beberapa kesalahan pembacaan, dimana, dari 30 kali pengujian, didapatkan 5 kesalahan pembacaan hasil sensor. Kesalahan ini sering terjadi pada range yang berdekatan dengan pembacaan sensor analog, contohnya di antara 3 dan 4 dimana sensor cukup sulit membedakan tanah kering ataupun agak basah, juga pada nilai 7 dan 8, dimana sensor sulit membedakan tanah agak basah ataupun tanah basah, sehingga terjadi kesalahan pembacaan. Dari percobaan di atas dapat dilihat bahwa alat ini memiliki persentase keberhasilan sekitar 83% dari 30 kali percobaan dalam membaca keadaan tanah, apakah itu kering, basah, maupun agak basah. Nilai ini bisa dikatakan cukup baik untuk menentukan seberapa baik alat dalam mendeteksi tanah yang basah, kering, maupun tanah

yang agak basah. Hasil pengujian juga dapat dilihat pada gambar 4.51.



Gambar 4.51 Hasil Pembacaan Sensor di dalam ruangan

Setelah melakukan percobaan di dalam ruangan, dilakukan juga percobaan di lahan, untuk mengetahui bagaimana pembacaan sensor terhadap tanah di siang hari, dan juga untuk melihat seberapa baik alat dalam mengontrol kelembapan. Hasil Percobaan untuk mendeteksi tanah dapat dilihat pada tabel 4.6.

Tabel 4.6 Pengujian sensor dalam mendeteksi kelembapan di lahan

Nilai Sensor Analog	Hasil Pembacaan Sensor
0	Kering
1	Kering
2	Kering
3	Kering
4	Kering
5	Agak Basah
6	Agak Basah
7	Basah
8	Basah

9	Basah
10	Basah

Dari tabel 4.6 dapat dilihat bahwa terdapat dua kali kesalahan pembacaan sensor, yaitu pada saat tanah yang seharusnya terbaca agak basah, namun tsensor membaca tanah kering, dan jga saat seharusnya tanah yang agak basah, namun terbaca basah. Hal ini dapat diakibatkan suhu tanah yang masih mendekati suhu saat kering maupun basah. Dari 10 percobaan didapatkan bahwa persentase keberhasilan dari pembacaan sensor adalah 80%. Hasil ini cukup baik untuk pendeteksian seberapa baik alat mendeteksi tanah kering, basah, maupun agak basah.



Gambar 4.52 Hasil Pembacaan Sensor di Lahan

Setelah pendeteksian, dilakukan pengujian seberapa baik alat dalam mengontrol kelembapan tanah, untuk mengetahui hal ini, dilakukan pengujian terhadap tanah yang kering, apakah alat menyiram sampai basah atau tidak. Pengujian ini hanya bisa dilakukan di lahan, karena jika penyiraman dilakukan di lahan, yang disiram adalah langsung tanah, namunjika penyiraman dilakukan di dalam ruangan, tanahnya tidak sebesar di lahan, maka dari itu, pengontrolan nilai kelembapan hanya dapat dilakukan dilahan, dan alat ini hanya dapat digunakan di lahan. Hasil percobaan seberapa baik alat mengontrol kelembapan dapat dilihat

pada tabel 4.7.

Tabel 4.7 Pengujian Pengontrolan kelembapan tanah

Nilai Awal Sensor Analog	Nilai Akhir Sensor Analog
0	10
1	10
2	10
3	10
4	10
5	10

Dari Tabel 4.7 dapat dilihat bahwa tanah yang kering langsung disiram sampai ke keadaan basah. Dapat dikatakan bahwa alat ini dapat mengontrol kelembapan tanah dari kering, sampai ke lembap dengan baik. Dari 6 kali percobaan, tanah dari keadaan kering dibasahi hingga mencapai keadaan basah dengan persentase keberhasilan 100%.



Gambar 4.54 Alat menyiram tanah yang kering

Hasil pembacaan sensor terhadap tanah setelah dilakukan penyiraman dapat dilihat padagambar 4.53 di bawah, dimana sensor menunjukkan bahwa tanah yang disiram telah sepenuhnya basah.



Gambar 4.54 Hasil Pembacaan Sensor Analog

.....*Halaman ini sengaja dikosongkan*.....

BAB V

PENUTUP

5.1. Kesimpulan

Pada Tugas Akhir ini, dibuatlah suatu sistem kontrol kelembapan tanah menggunakan kamera termal dan fuzzy logic, dimana sistem ini menggunakan kamera termal sebagai sensor pengganti sensor kelembapan, kemudian juga digunakan DS3231 untuk menentukan waktu, dimana ini dibutuhkan untuk penentuan waktu penyiraman. Kemudian sistem ini menggunakan kontrol fuzzy logic, dimana kontrol fuzzy logic digunakan untuk mengontrol lama bukaan valve yang digunakan. Kemudian valve yang digunakan adalah solenoid valve, dimana valve ini akan terbuka jika dialiri listrik.

Berdasarkan percobaan yang telah dilakukan pada pelaksanaan tugas akhir ini didapat beberapa kesimpulan sebagai berikut:

Pada kondisi yang sama, namun di waktu yang berbeda, hasil pembacaan sensor juga berbeda. Hal ini dikarenakan adanya perbedaan suhu antara pagi, siang, maupun sore, sehingga mempengaruhi pembacaan sensor pada permukaan tanah. Suhu maksimum terjadi pada saat siang hari, sehingga penyiraman hanya akan dilakukan sekali saja, yaitu pada siang hari.

Semakin besar kelembapan pada tanah, maka suhu yang terdeteksi pada sensor AMG8833 semakin kecil.

Penggunaan suhu air yang berbeda dapat mempengaruhi hasil pembacaan sensor, meskipun tanah dalam keadaan basah, namun jika air yang digunakan untuk menyiram tanah adalah air panas, maka hasil pembacaan sensor menunjukkan bahwa tanah kering, karena hasil pembacaan sensor sangat besar.

5.2. Saran

Sebagai sarana pengembangan Tugas Akhir ini, maka terdapat beberapa saran dari penulis berdasarkan hasil yang diperoleh saat percobaan, yaitu sebagai berikut:

Untuk penyiraman, dapat dilakukan dengan pompa yang bertekanan kuat, kemudian diberi penyebar air pada ujung selangnya, sehingga air bisa menyebar dan tidak terkumpul di satu titik.

.....*Halaman ini sengaja dikosongkan*.....

DAFTAR PUSTAKA

- [1] “What is soil? | COSMOS-UK.” [Online]. Available: <https://cosmos.ceh.ac.uk/soil>. [Accessed: 13-Nov-2018].
- [2] “(PDF) Analisis Kelembaban Tanah dengan Citra Landsat 7 ETM+.” [Online]. Available: https://www.researchgate.net/publication/306284643_Analisis_Kelembaban_Tanah_dengan_Citra_Landsat_7_ETM. [Accessed: 12-Nov-2018].
- [3] Earthzine, “Why We Should Start Thinking About Soil Moisture,” *IEEE Earthzine*, 03-Feb-2015. .
- [4] “Why measure soil moisture? | COSMOS-UK.” [Online]. Available: <https://cosmos.ceh.ac.uk/whymeasuresoilmoisture>. [Accessed: 13-Nov-2018].
- [5] “What Are Soils? | Learn Science at Scitable.” [Online]. Available: <https://www.nature.com/scitable/knowledge/library/what-are-soils-67647639>. [Accessed: 18-Nov-2018].
- [6] “Study of Physical Properties of Soil (Theory) : Class 12 : Biology : Amrita Online Lab.” [Online]. Available: <http://amrita.olabs.edu.in/?brch=18&cnt=1&sim=235&sub=79>. [Accessed: 18-Nov-2018].
- [7] admin, “All About Soil,” *Co+op, stronger together*, 06-Oct-2010. [Online]. Available: <https://www.strongertogether.coop/fresh-from-the-source/growing-your-own-food/all-about-soil>. [Accessed: 20-Nov-2018].
- [8] “How to cultivate in different soil structures - Väderstad.” [Online]. Available: <https://www.vaderstad.com/en/know-how/basic-agronomy/soil-basics/soil-structure/>. [Accessed: 22-Nov-2018].
- [9] “terraGIS Soil.” [Online]. Available: http://www.terragis.bees.unsw.edu.au/terraGIS_soil/sp_water-soil_moisture_classification.html. [Accessed: 10-Apr-2019].
- [10] “Blackbody Radiation.” [Online]. Available: <http://hyperphysics.phy-astr.gsu.edu/hbase/mod6.html>. [Accessed: 22-Nov-2018].
- [11] “Thermal radiation,” *Wikipedia*. 21-Nov-2018.
- [12] “Blackbody Radiation | Astronomy 801: Planets, Stars, Galaxies, and the Universe.” [Online]. Available: https://www.e-education.psu.edu/astro801/content/13_p5.html. [Accessed: 22-Nov-2018].

- [13] “Infrared Waves | Science Mission Directorate.” [Online]. Available: https://science.nasa.gov/ems/07_infraredwaves. [Accessed: 23-Nov-2018].
- [14] “What is infrared radiation (IR)? - Definition from WhatIs.com,” *SearchNetworking*. [Online]. Available: <https://searchnetworking.techtarget.com/definition/infrared-radiation>. [Accessed: 23-Nov-2018].
- [15] “Infrared Spectroscopy.” [Online]. Available: <https://www2.chemistry.msu.edu/faculty/reusch/virttxtjml/spectrpy/infrared/infrared.htm>. [Accessed: 10-Apr-2019].
- [16] “Top Uses and Applications of Thermal Imaging Cameras - Quick Tips #345 - Grainger Industrial Supply.” [Online]. Available: <https://www.grainger.com/content/qt-thermal-imaging-applications-uses-features-345>. [Accessed: 22-Nov-2018].
- [17] V. O. M. T. POSTED 02/16/2016, “What is Thermal Imaging?,” *Vision Online*. [Online]. Available: <https://www.visiononline.org/blog-article.cfm/What-is-Thermal-Imaging/8>. [Accessed: 22-Nov-2018].
- [18] “Overview | Adafruit AMG8833 8x8 Thermal Camera Sensor | Adafruit Learning System.” [Online]. Available: <https://learn.adafruit.com/adafruit-amg8833-8x8-thermal-camera-sensor>. [Accessed: 23-Nov-2018].
- [19] “Arduino Uno R3 | ATmega328P Microcontroller Board,” *Module143*. [Online]. Available: <https://www.module143.com/arduino-uno-r3-atmega328p-microcontroller-board>. [Accessed: 11-Apr-2019].
- [20] “Arduino Uno Rev3 - Education - Arduino.” [Online]. Available: <https://store.arduino.cc/usa/arduino-uno-rev3>. [Accessed: 11-Apr-2019].
- [21] “Arduino - Board.” [Online]. Available: <https://www.arduino.cc/en/reference/board>. [Accessed: 11-Apr-2019].
- [22] “Arduino Uno Rev3 SMD.” [Online]. Available: <https://shop.evilmadscientist.com/productsmenu/564>. [Accessed: 11-Apr-2019].
- [23] “Arduino Reference.” [Online]. Available: <https://www.arduino.cc/reference/en/language/functions/communication/serial/>. [Accessed: 11-Apr-2019].
- [24] “Arduino Uno For Beginners - Projects, Programming and Parts

- (Tutorial),” *Makerspaces.com*, 07-Feb-2017. .
- [25] R. Farana, B. Walek, M. Janošek, and J. Žáček, “Fuzzy-Logic Control in fast technological processes,” in *Proceedings of the 2014 15th International Carpathian Control Conference (ICCC)*, 2014, pp. 105–108.
- [26] “What is ‘fuzzy logic’? Are there computers that are inherently fuzzy and do not apply the usual binary logic?,” *Scientific American*. [Online]. Available: <https://www.scientificamerican.com/article/what-is-fuzzy-logic-are-t/>. [Accessed: 22-Nov-2018].
- [27] “(PDF) Rancang Bangun Sistem Irigasi Tanaman Otomatis Menggunakan Wireless Sensor Network.” [Online]. Available: https://www.researchgate.net/publication/318561340_Rancang_Bangun_Sistem_Irigasi_Tanaman_Otomatis_Menggunakan_Wireless_Sensor_Network. [Accessed: 12-Nov-2018].
- [28] “Fuzzy Logic Tutorial: What is, Application & Example.” [Online]. Available: <https://www.guru99.com/what-is-fuzzy-logic.html>. [Accessed: 22-May-2019].
- [29] tutorialspoint.com, “Artificial Intelligence Fuzzy Logic Systems,” *www.tutorialspoint.com*. [Online]. Available: https://www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_fuzzy_logic_systems.htm. [Accessed: 22-May-2019].
- [30] 081211332008 FADLY DWI KARDIANTO, “RANCANG BANGUN MONITORING TEMPERATUR BERBASIS ARDUINO UNO UNTUK PENGUKURAN TEMPERATUR PADA EKSPERIMEN TARA KALOR MEKANIK,” skripsi, Universitas Airlangga, 2017.
- [31] “1602 16x2 Blue LCD Module,” *ElectroDragon*. .
- [32] D. Oktanugraha, “PERANCANGAN ANTARMUKA I2C PADA SENSOR CO2 MHZ-19,” May 2018.
- [33] “Bekerja dengan I2C LCD dan Arduino – Saptaji.com.” [Online]. Available: <http://saptaji.com/2016/06/27/bekerja-dengan-i2c-lcd-dan-arduino/>. [Accessed: 01-Dec-2018].
- [34] “Arduino DS3231 RTC Module Tutorial - Real Time Clock,” *Electronics Hub*, 29-Aug-2018. .
- [35] “HC-05 Bluetooth to Serial Module,” 13-May-2012. .
- [36] “File:FC-114.jpg - ePro Labs WiKi.” [Online]. Available: <https://wiki.eprolabs.com/index.php?title=File:FC-114.jpg>. [Accessed: 24-Jun-2019].

- [37] “Jual Solenoid Valve Plastik DC12V - Kota Surabaya - akhi_shop,” *Tokopedia*. [Online]. Available: <https://www.tokopedia.com/akhishop/solenoid-valve-plastik-dc12v>. [Accessed: 24-Jun-2019].
- [38] “Cara kerja solenoid valve.” [Online]. Available: <http://www.kitomaindonesia.com/article/19/cara-kerja-solenoid-valve>. [Accessed: 24-Jun-2019].
- [39] G. L. Importantbook, “GO LOOK IMPORTANTBOOK: solenoid valve equations in electronics circuits as like as Electrolytic Capacitor in Solenoid Circuit combine to transistors and relay electronics AMNIMARJESLOW GOVERNMENT 91220017 XI XA PIN PING HUNG CHOP 02096010014 LJBUSAF IF THEN GO TO CONTROL INSTRUMENT SOLENOID VALVE COMBINE TRANSISTORS AND THEN ACTIVE COMPONENT IN ELECTRONICS FORWARD STABILIZATION FLOWING MEDIUM 2020,” *GO LOOK IMPORTANTBOOK*, 12-Mar-2018. .
- [40] R. Gemilang, “Pengertian Relay, Fungsi, Dan Cara Kerja Relay,” *Immersa Lab*, 02-Mar-2018. .
- [41] D. Kho, “Pengertian Relay dan Fungsi Relay,” *Teknik Elektronika*, 07-Mar-2015. .

LAMPIRAN A

Program Pembacaan Sensor dan Sistem Fuzzy Logic

Program Arduino Kontrol Kelembapan Tanah Menggunakan Kamera Termal dan Fuzzy Logic

```
#include <Wire.h>
#include <LoRaLib.h>
#include <LiquidCrystal_I2C.h>
#include <FuzzyRule.h>
#include <FuzzyComposition.h>
#include <Fuzzy.h>
#include <FuzzyRuleConsequent.h>
#include <FuzzyOutput.h>
#include <FuzzyInput.h>
#include <FuzzyIO.h>
#include <FuzzySet.h>
#include <FuzzyRuleAntecedent.h>
#include <Wire.h>
#include <Adafruit_AMG88xx.h>
#include <DS3231.h>
#include <SoftwareSerial.h>
```

```
Adafruit_AMG88xx amg;
DS3231 rtc(SDA,SCL);
Time waktu;
SoftwareSerial blue(12,13);
```

```
float error,error,last_error,pwm,output1;
Fuzzy* fuzzy = new Fuzzy();
```

```
float pixels[AMG88xx_PIXEL_ARRAY_SIZE];
float ave;
```

```
FuzzySet* no = new FuzzySet(0, 0, 0, 0);
FuzzySet* sedikit = new FuzzySet(0, 10, 10, 20);
FuzzySet* sedang = new FuzzySet(10, 20, 20, 30);
FuzzySet* banyak = new FuzzySet(20, 30, 30, 35);
```

```
void setup() {
```

```

Wire.begin();
Serial.begin(38400);
blue.begin(38400);
rtc.begin();

bool status;

status = amg.begin();
if (!status) {
    Serial.println("Could not find a valid AMG88xx sensor, check
wiring!");
    while (1);
}
delay(100);
}

void Pagi(){
    Pagi();
    display_Pagi();
}

void Siang(){
    Siang();
    display_siang;
}

void Sore(){
    Sore();
    display_sore;
}

void BacaAMGRTC(){
    amg.readPixels(pixels);
    Serial.print("Grid-EYE:\r\n");
    for(int i=1; i<=AMG88xx_PIXEL_ARRAY_SIZE; i++){
        Serial.print(pixels[i-1]);
        Serial.print(" ");
        if( i%8 == 0 ) Serial.println();
    }
}

```

```

float sum = 0;
for(int i=0; i<64; i++){
    sum = sum + pixels[i];
}
ave = sum/64;
Serial.println();

Serial.print("Rata-rata = ");
Serial.println(ave);

Serial.println(rtc.getDOWStr(FORMAT_LONG));

waktu = rtc.getTime();
int dataJam = waktu.hour;
int dataMenit = waktu.min;
int dataDetik = waktu.sec;
Serial.print(dataJam);
Serial.print(':');
Serial.print(dataMenit);
Serial.print(':');
Serial.println(dataDetik);

delay(1000);
}

void loop() {
    BacaAMGRTC();
    waktu = rtc.getTime();
    int dataJam = waktu.hour;
    if(dataJam >= 0 && dataJam <=11){
        Pagi();
    }
    else if(dataJam > 11 && dataJam <= 14){
        Siang();
    }
    else if(dataJam >= 15 && dataJam < 24){
        Sore();
    }
}

```

```
float setpoint3=24;
```

```
FuzzySet* error6 = new FuzzySet(-15, -15, -10, 10);  
FuzzySet* error7 = new FuzzySet(-10, 10, 10, 30);  
FuzzySet* error8 = new FuzzySet(10, 30, 30, 50);  
FuzzySet* error9 = new FuzzySet(40, 60, 85, 85);  
//Input Derror  
FuzzySet* derror6 = new FuzzySet(-20, -20, -1, 3);  
FuzzySet* derror7 = new FuzzySet(-1, 3, 3, 7);  
FuzzySet* derror8 = new FuzzySet(5, 7, 30, 30);  
FuzzySet* derror9 = new FuzzySet(5, 7, 30, 30);
```

```
void Pagi(){  
    // Membuat fuzzyinput  
    //Fuzzy Input error  
    FuzzyInput* inputerror = new FuzzyInput(1);  
    inputerror->addFuzzySet(error6);  
    inputerror->addFuzzySet(error7);  
    inputerror->addFuzzySet(error8);  
    inputerror->addFuzzySet(error9);  
    fuzzy->addFuzzyInput(inputerror);  
    //Fuzzy Input Derror  
    FuzzyInput* inputDerror = new FuzzyInput(2);  
    inputDerror->addFuzzySet(derror6);  
    inputDerror->addFuzzySet(derror7);  
    inputDerror->addFuzzySet(derror8);  
    inputDerror->addFuzzySet(derror9);  
    fuzzy->addFuzzyInput(inputDerror);  
    //Step 4. Membuat Fuzzy Rule  
    //1. IF error=error6 and Derror=derror6 THEN velocity=stop  
    FuzzyRuleAntecedent* iferrorerror6AndDerrorderror6 = new  
    FuzzyRuleAntecedent();  
    iferrorerror6AndDerrorderror6->joinWithAND(error6, derror6);  
    FuzzyRuleConsequent* thenVelocityStop14 = new  
    FuzzyRuleConsequent(); // Instantiating a Consequent to expression  
    thenVelocityStop14->addOutput(no);// Adding corresponding  
    FuzzySet to Consequent object  
    // Instantiating a FuzzyRule object  
    FuzzyRule* fuzzyRule14 = new FuzzyRule(14,
```

```

iferrorerror6AndDerrorderror6, thenVelocityStop14); // Passing the
Antecedent and the Consequent of expression
fuzzy->addFuzzyRule(fuzzyRule14); // Adding FuzzyRule to Fuzzy
object

```

```

//2. IF error=N and Derror=negkec THEN velocity=stop
FuzzyRuleAntecedent* iferrorerror7AndDerrorderror6 = new
FuzzyRuleAntecedent();
iferrorerror7AndDerrorderror6->joinWithAND(error7, derror6);
FuzzyRuleConsequent* thenVelocityStop15 = new
FuzzyRuleConsequent();
thenVelocityStop15->addOutput(no);
// Instantiating a FuzzyRule object
FuzzyRule* fuzzyRule15 = new FuzzyRule(15,
iferrorerror7AndDerrorderror6, thenVelocityStop15);
fuzzy->addFuzzyRule(fuzzyRule15);

```

```

//3. IF error=Z and Derror=negkec THEN velocity=slow
FuzzyRuleAntecedent* iferrorerror8AndDerrorderror6 = new
FuzzyRuleAntecedent();
iferrorerror8AndDerrorderror6->joinWithAND(error8, derror6);
FuzzyRuleConsequent* thenVelocitySlow16 = new
FuzzyRuleConsequent();
thenVelocitySlow16->addOutput(sedikit);
// Instantiating a FuzzyRule object
FuzzyRule* fuzzyRule16 = new FuzzyRule(16,
iferrorerror8AndDerrorderror6, thenVelocitySlow16);
fuzzy->addFuzzyRule(fuzzyRule16);

```

```

//3. IF error=Z and Derror=negkec THEN velocity=slow
FuzzyRuleAntecedent* iferrorerror9AndDerrorderror6 = new
FuzzyRuleAntecedent();
iferrorerror9AndDerrorderror6->joinWithAND(error9, derror6);
FuzzyRuleConsequent* thenVelocitySlow17 = new
FuzzyRuleConsequent();
thenVelocitySlow17->addOutput(sedikit);
// Instantiating a FuzzyRule object
FuzzyRule* fuzzyRule17 = new FuzzyRule(17,
iferrorerror8AndDerrorderror6, thenVelocitySlow17);

```

```

fuzzy->addFuzzyRule(fuzzyRule17);

//1. IF error=NK and Derror=negkec THEN velocity=stop
FuzzyRuleAntecedent* iferrorerror6AndDerrorerror7 = new
FuzzyRuleAntecedent();
iferrorerror6AndDerrorerror6->joinWithAND(error6, derror7);
FuzzyRuleConsequent* thenVelocityStop18 = new
FuzzyRuleConsequent(); // Instantiating a Consequent to expression
thenVelocityStop18->addOutput(no); // Adding corresponding
FuzzySet to Consequent object
// Instantiating a FuzzyRule object
FuzzyRule* fuzzyRule18 = new FuzzyRule(18,
iferrorerror6AndDerrorerror7, thenVelocityStop18); // Passing the
Antecedent and the Consequent of expression
fuzzy->addFuzzyRule(fuzzyRule18); // Adding FuzzyRule to Fuzzy
object

//2. IF error=N and Derror=negkec THEN velocity=stop
FuzzyRuleAntecedent* iferrorerror7AndDerrorerror7 = new
FuzzyRuleAntecedent();
iferrorerror7AndDerrorerror7->joinWithAND(error7, derror7);
FuzzyRuleConsequent* thenVelocityStop19 = new
FuzzyRuleConsequent();
thenVelocityStop19->addOutput(sedikit);
// Instantiating a FuzzyRule object
FuzzyRule* fuzzyRule19 = new FuzzyRule(19,
iferrorerror7AndDerrorerror7, thenVelocityStop19);
fuzzy->addFuzzyRule(fuzzyRule19);

//3. IF error=Z and Derror=negkec THEN velocity=slow
FuzzyRuleAntecedent* iferrorerror8AndDerrorerror7 = new
FuzzyRuleAntecedent();
iferrorerror8AndDerrorerror7->joinWithAND(error8, derror7);
FuzzyRuleConsequent* thenVelocitySlow20 = new
FuzzyRuleConsequent();
thenVelocitySlow20->addOutput(sedang);
// Instantiating a FuzzyRule object
FuzzyRule* fuzzyRule20 = new FuzzyRule(20,
iferrorerror8AndDerrorerror7, thenVelocitySlow20);

```

```

fuzzy->addFuzzyRule(fuzzyRule20);

//3. IF error=Z and Derror=negkec THEN velocity=slow
FuzzyRuleAntecedent* iferrorerror9AndDerrorerror7 = new
FuzzyRuleAntecedent();
iferrorerror9AndDerrorerror7->joinWithAND(error9, derror7);
FuzzyRuleConsequent* thenVelocitySlow21 = new
FuzzyRuleConsequent();
thenVelocitySlow21->addOutput(sedang);
// Instantiating a FuzzyRule object
FuzzyRule* fuzzyRule21 = new FuzzyRule(21,
iferrorerror8AndDerrorerror7, thenVelocitySlow21);
fuzzy->addFuzzyRule(fuzzyRule21);

//1. IF error=NK and Derror=negkec THEN velocity=stop
FuzzyRuleAntecedent* iferrorerror6AndDerrorerror8 = new
FuzzyRuleAntecedent();
iferrorerror6AndDerrorerror8->joinWithAND(error6, derror8);
FuzzyRuleConsequent* thenVelocityStop22 = new
FuzzyRuleConsequent(); // Instantiating a Consequent to expression
thenVelocityStop22->addOutput(no); // Adding corresponding
FuzzySet to Consequent object
// Instantiating a FuzzyRule object
FuzzyRule* fuzzyRule22 = new FuzzyRule(22,
iferrorerror6AndDerrorerror8, thenVelocityStop22); // Passing the
Antecedent and the Consequent of expression
fuzzy->addFuzzyRule(fuzzyRule22); // Adding FuzzyRule to Fuzzy
object

//2. IF error=N and Derror=negkec THEN velocity=stop
FuzzyRuleAntecedent* iferrorerror7AndDerrorerror8 = new
FuzzyRuleAntecedent();
iferrorerror7AndDerrorerror8->joinWithAND(error7, derror8);
FuzzyRuleConsequent* thenVelocityStop23 = new
FuzzyRuleConsequent();
thenVelocityStop23->addOutput(sedikit);
// Instantiating a FuzzyRule object
FuzzyRule* fuzzyRule23 = new FuzzyRule(23,
iferrorerror7AndDerrorerror8, thenVelocityStop23);

```

```

fuzzy->addFuzzyRule(fuzzyRule23);

//3. IF error=Z and Derror=negkec THEN velocity=slow
FuzzyRuleAntecedent* iferrorerror8AndDerrorerror8 = new
FuzzyRuleAntecedent();
iferrorerror8AndDerrorerror8->joinWithAND(error8, derror8);
FuzzyRuleConsequent* thenVelocitySlow24 = new
FuzzyRuleConsequent();
thenVelocitySlow24->addOutput(sedang);
// Instantiating a FuzzyRule object
FuzzyRule* fuzzyRule24 = new FuzzyRule(24,
iferrorerror8AndDerrorerror8, thenVelocitySlow24);
fuzzy->addFuzzyRule(fuzzyRule24);

//3. IF error=Z and Derror=negkec THEN velocity=slow
FuzzyRuleAntecedent* iferrorerror9AndDerrorerror8 = new
FuzzyRuleAntecedent();
iferrorerror9AndDerrorerror8->joinWithAND(error9, derror8);
FuzzyRuleConsequent* thenVelocitySlow25 = new
FuzzyRuleConsequent();
thenVelocitySlow25->addOutput(banyak);
// Instantiating a FuzzyRule object
FuzzyRule* fuzzyRule25 = new FuzzyRule(25,
iferrorerror8AndDerrorerror8, thenVelocitySlow25);
fuzzy->addFuzzyRule(fuzzyRule25);

//1. IF error=NK and Derror=negkec THEN velocity=stop
FuzzyRuleAntecedent* iferrorerror6AndDerrorerror9 = new
FuzzyRuleAntecedent();
iferrorerror6AndDerrorerror9->joinWithAND(error6, derror9);
FuzzyRuleConsequent* thenVelocityStop26 = new
FuzzyRuleConsequent(); // Instantiating a Consequent to expression
thenVelocityStop26->addOutput(sedang); // Adding corresponding
FuzzySet to Consequent object
// Instantiating a FuzzyRule object
FuzzyRule* fuzzyRule26 = new FuzzyRule(26,
iferrorerror6AndDerrorerror9, thenVelocityStop26); // Passing the
Antecedent and the Consequent of expression
fuzzy->addFuzzyRule(fuzzyRule26); // Adding FuzzyRule to Fuzzy

```


object

```
//2. IF error=N and Derror=negkec THEN velocity=stop
FuzzyRuleAntecedent* iferrorerror7AndDerrorerror9 = new
FuzzyRuleAntecedent();
iferrorerror7AndDerrorerror9->joinWithAND(error7, derror9);
FuzzyRuleConsequent* thenVelocityStop27 = new
FuzzyRuleConsequent();
thenVelocityStop27->addOutput(banyak);
// Instantiating a FuzzyRule object
FuzzyRule* fuzzyRule27 = new FuzzyRule(27,
iferrorerror7AndDerrorerror9, thenVelocityStop27);
fuzzy->addFuzzyRule(fuzzyRule27);

//3. IF error=Z and Derror=negkec THEN velocity=slow
FuzzyRuleAntecedent* iferrorerror8AndDerrorerror9 = new
FuzzyRuleAntecedent();
iferrorerror8AndDerrorerror9->joinWithAND(error8, derror9);
FuzzyRuleConsequent* thenVelocitySlow28 = new
FuzzyRuleConsequent();
thenVelocitySlow28->addOutput(banyak);
// Instantiating a FuzzyRule object
FuzzyRule* fuzzyRule28 = new FuzzyRule(28,
iferrorerror8AndDerrorerror9, thenVelocitySlow28);
fuzzy->addFuzzyRule(fuzzyRule28);

//3. IF error=Z and Derror=negkec THEN velocity=slow
FuzzyRuleAntecedent* iferrorerror9AndDerrorerror9 = new
FuzzyRuleAntecedent();
iferrorerror9AndDerrorerror9->joinWithAND(error9, derror9);
FuzzyRuleConsequent* thenVelocitySlow29 = new
FuzzyRuleConsequent();
thenVelocitySlow29->addOutput(banyak);
// Instantiating a FuzzyRule object
FuzzyRule* fuzzyRule29 = new FuzzyRule(29,
iferrorerror8AndDerrorerror9, thenVelocitySlow29);
fuzzy->addFuzzyRule(fuzzyRule29);
}
```

```

void display_Pagi(){

    /*Controlled with fuzzy logic controller and gas sensors*/
    last_error = error;
    error = setpoint3 - ave ;
    derror = error - last_error;

    Serial.println(error);
    Serial.println(derror);

    fuzzy->setInput(5, error);
    fuzzy->setInput(6, derror);
    fuzzy->fuzzify();
    //Tampilkan data Error
    Serial.print("Error: ");
    Serial.print(error6->getPertinence());    Serial.print(", ");
    Serial.print(error7->getPertinence());    Serial.print(", ");
    Serial.println(error8->getPertinence());    Serial.print(", ");
    //Tampilkan data Derror
    Serial.print("Derror: ");
    Serial.print(derror6->getPertinence());Serial.print(", ");
    Serial.print(derror7->getPertinence());    Serial.print(", ");
    Serial.println(derror8->getPertinence());    Serial.print(", ");

    output1 = fuzzy->defuzzify(3);
    Serial.print("KELUARAN AVERAGE: ");    Serial.println(output1);

    if(output1 >= +4){
        blue.write("A");
        delay(8000);
    }
    else if(output1 > 2 && output < +4){
        blue.write("B");
        delay(4000);
    }
    else if(output1 <= 2){
        blue.write("C");
        delay(2000);
    }
}

```

```

    }
}
float setpoint2=36;
/*****FUZZY LOGIC
CONTROLLER*****/
*****/
// Step 1. Inisialisasi object library

//Input error
//misal error=57-70=13, derror=0-13
FuzzySet* error1 = new FuzzySet(-70, -70, -10, 20);
FuzzySet* error2 = new FuzzySet(-10, 20, 30, 30);
//Input Derror
FuzzySet* derror1 = new FuzzySet(-35, -35, -2, 4);
FuzzySet* derror2 = new FuzzySet(-2, 4, 15, 15);

void Siang()
{
    FuzzyInput* inputerror = new FuzzyInput(3);
    inputerror->addFuzzySet(error1);
    inputerror->addFuzzySet(error2);
    fuzzy->addFuzzyInput(inputerror);
    //Fuzzy Input Derror
    FuzzyInput* inputDerror = new FuzzyInput(4);
    inputDerror->addFuzzySet(derror1);
    inputDerror->addFuzzySet(derror2);
    fuzzy->addFuzzyInput(inputDerror);
    // Step 3. membuat fuzzy output
    FuzzyOutput* outputSpeed = new FuzzyOutput(1);
    outputSpeed->addFuzzySet(no);
    outputSpeed->addFuzzySet(sedikit);
    outputSpeed->addFuzzySet(sedang);
    outputSpeed->addFuzzySet(banyak);
    fuzzy->addFuzzyOutput(outputSpeed);

    //Step 4. Membuat Fuzzy Rule
    //1. IF error=error1 and Derror=derror1 THEN velocity=stop
    FuzzyRuleAntecedent* iferrorerror1AndDerrorderror1 = new

```

```

FuzzyRuleAntecedent();
  iferrorerror1AndDerrorderror1->joinWithAND(error1, derror1);
  FuzzyRuleConsequent*   thenVelocityStop10   =   new
FuzzyRuleConsequent(); // Instantiating a Consequent to expression
  thenVelocityStop10->addOutput(no); // Adding corresponding
FuzzySet to Consequent object
  // Instantiating a FuzzyRule object
  FuzzyRule*   fuzzyRule10   =   new   FuzzyRule(10,
iferrorerror1AndDerrorderror1, thenVelocityStop10); // Passing the
Antecedent and the Consequent of expression
  fuzzy->addFuzzyRule(fuzzyRule10); // Adding FuzzyRule to Fuzzy
object

```

```

//2. IF error=N and Derror=derror1 THEN velocity=stop
FuzzyRuleAntecedent*   iferrorerror2AndDerrorderror1   =   new
FuzzyRuleAntecedent();
  iferrorerror2AndDerrorderror1->joinWithAND(error2, derror1);
  FuzzyRuleConsequent*   thenVelocityStop11   =   new
FuzzyRuleConsequent();
  thenVelocityStop11->addOutput(no);
  // Instantiating a FuzzyRule object
  FuzzyRule*   fuzzyRule11   =   new   FuzzyRule(11,
iferrorerror2AndDerrorderror1, thenVelocityStop11);
  fuzzy->addFuzzyRule(fuzzyRule11);

```

```

//6. IF error=error1 and Derror=derror2 THEN velocity=stop
FuzzyRuleAntecedent*   iferrorerror1AndDerrorderror2   =   new
FuzzyRuleAntecedent();
  iferrorerror1AndDerrorderror2->joinWithAND(error1, derror2);
  FuzzyRuleConsequent*   thenVelocityStop12   =   new
FuzzyRuleConsequent();
  thenVelocityStop12->addOutput(no);
  // Instantiating a FuzzyRule object
  FuzzyRule*   fuzzyRule12   =   new   FuzzyRule(12,
iferrorerror1AndDerrorderror2, thenVelocityStop12);
  fuzzy->addFuzzyRule(fuzzyRule12);

```

```

//7. IF error=N and Derror=derror2 THEN velocity=slow
FuzzyRuleAntecedent*   iferrorerror2AndDerrorderror2   =   new

```

```

FuzzyRuleAntecedent();
  iferrorerror2AndDerrorderror2->joinWithAND(error2, derror2);
  FuzzyRuleConsequent*      thenVelocitySlow13      =      new
FuzzyRuleConsequent();
  thenVelocitySlow13->addOutput(sedik);
  // Instantiating a FuzzyRule object
  FuzzyRule*      fuzzyRule13      =      new      FuzzyRule(13,
iferrorerror2AndDerrorderror2, thenVelocitySlow13);
  fuzzy->addFuzzyRule(fuzzyRule13);

}

void display_siang()
{
  /*Controlled with fuzzy logic controller and gas sensors*/
  last_error = error;
  error = setpoint2 - ave;
  derror = error - last_error;

  Serial.println(error);
  Serial.println(derror);

  fuzzy->setInput(3, error);
  fuzzy->setInput(4, derror);
  fuzzy->fuzzify();
  //Tampilkan data Error
  Serial.print("Error: ");
  Serial.print(error1->getPertinence());      Serial.print(", ");
  Serial.print(error2->getPertinence());      Serial.print(", ");
  //Tampilkan data Derror
  Serial.print("Derror: ");
  Serial.print(derror1->getPertinence());Serial.print(", ");
  Serial.print(derror2->getPertinence());      Serial.print(", ");

  output1 = fuzzy->defuzzify(2);
  Serial.print("KELUARAN AVERAGE: ");      Serial.println(output1);
  if(output1 >= +4){
    blue.write("D");
    delay(8000);
  }
}

```

```

    }
    else if(output1>2 && output1<+4){
        blue.write("E");
        delay(4000);
    }
    else if(output1<=2){
        blue.write("F");
        delay(2000);
    }
}
float setpoint1=22;

// Step 1. Inisialisasi object library

//Input error
//misal error=57-70=13, derror=0-13
FuzzySet* error3 = new FuzzySet(-40, -40, -5, 15);
FuzzySet* error4 = new FuzzySet(-5, 15, 15, 35);
FuzzySet* error5 = new FuzzySet(15, 35, 60, 60);
//Input Derror
FuzzySet* derror3 = new FuzzySet(-20, -20, -1, 3);
FuzzySet* derror4 = new FuzzySet(-1, 3, 3, 7);
FuzzySet* derror5 = new FuzzySet(5, 7, 30, 30);
//OutputSpeed

void Sore(){
    //Fuzzy Input error
    FuzzyInput* inputerror = new FuzzyInput(1);
    inputerror->addFuzzySet(error3);
    inputerror->addFuzzySet(error4);
    inputerror->addFuzzySet(error5);
    fuzzy->addFuzzyInput(inputerror);
    //Fuzzy Input Derror
    FuzzyInput* inputDerror = new FuzzyInput(2);
    inputDerror->addFuzzySet(derror3);
    inputDerror->addFuzzySet(derror4);
    inputDerror->addFuzzySet(derror5);
    fuzzy->addFuzzyInput(inputDerror);
}

```

```

// Step 3. membuat fuzzy output
FuzzyOutput* outputSpeed = new FuzzyOutput(1);
outputSpeed->addFuzzySet(no);
outputSpeed->addFuzzySet(sedikit);
outputSpeed->addFuzzySet(sedang);
outputSpeed->addFuzzySet(banyak);
fuzzy->addFuzzyOutput(outputSpeed);

//Step 4. Membuat Fuzzy Rule
//1. IF error=error3 and Derror=derror3 THEN velocity=stop
FuzzyRuleAntecedent* iferrorerror3AndDerrorerror3 = new
FuzzyRuleAntecedent();
iferrorerror3AndDerrorerror3->joinWithAND(error3, derror3);
FuzzyRuleConsequent* thenVelocityStop1 = new
FuzzyRuleConsequent(); // Instantiating a Consequent to expression
thenVelocityStop1->addOutput(no); // Adding corresponding FuzzySet
to Consequent object
// Instantiating a FuzzyRule object
FuzzyRule* fuzzyRule01 = new FuzzyRule(1,
iferrorerror3AndDerrorerror3, thenVelocityStop1); // Passing the
Antecedent and the Consequent of expression
fuzzy->addFuzzyRule(fuzzyRule01); // Adding FuzzyRule to Fuzzy
object

//2. IF error=error4 and Derror=derror3 THEN velocity=stop
FuzzyRuleAntecedent* iferrorerror4AndDerrorerror3 = new
FuzzyRuleAntecedent();
iferrorerror4AndDerrorerror3->joinWithAND(error4, derror3);
FuzzyRuleConsequent* thenVelocityStop2 = new
FuzzyRuleConsequent();
thenVelocityStop2->addOutput(no);
// Instantiating a FuzzyRule object
FuzzyRule* fuzzyRule02 = new FuzzyRule(2,
iferrorerror4AndDerrorerror3, thenVelocityStop2);
fuzzy->addFuzzyRule(fuzzyRule02);

//3. IF error=error5 and Derror=derror3 THEN velocity=slow
FuzzyRuleAntecedent* iferrorerror5AndDerrorerror3 = new
FuzzyRuleAntecedent();

```

```

iferrorerror5AndDerrorderror3->joinWithAND(error5, derror3);
FuzzyRuleConsequent* thenVelocitySlow3 = new
FuzzyRuleConsequent();
thenVelocitySlow3->addOutput(no);
// Instantiating a FuzzyRule object
FuzzyRule* fuzzyRule03 = new FuzzyRule(3,
iferrorerror5AndDerrorderror3, thenVelocitySlow3);
fuzzy->addFuzzyRule(fuzzyRule03);

//6. IF error=error3 and Derror=derror4 THEN velocity=stop
FuzzyRuleAntecedent* iferrorerror3AndDerrorderror4 = new
FuzzyRuleAntecedent();
iferrorerror3AndDerrorderror4->joinWithAND(error3, derror4);
FuzzyRuleConsequent* thenVelocityStop4 = new
FuzzyRuleConsequent();
thenVelocityStop4->addOutput(no);
// Instantiating a FuzzyRule object
FuzzyRule* fuzzyRule04 = new FuzzyRule(4,
iferrorerror3AndDerrorderror4, thenVelocityStop4);
fuzzy->addFuzzyRule(fuzzyRule04);
//7. IF error=error4 and Derror=derror4 THEN velocity=slow
FuzzyRuleAntecedent* iferrorerror4AndDerrorderror4 = new
FuzzyRuleAntecedent();
iferrorerror4AndDerrorderror4->joinWithAND(error4, derror4);
FuzzyRuleConsequent* thenVelocitySlow5 = new
FuzzyRuleConsequent();
thenVelocitySlow5->addOutput(sedikit);
// Instantiating a FuzzyRule object
FuzzyRule* fuzzyRule05 = new FuzzyRule(5,
iferrorerror4AndDerrorderror4, thenVelocitySlow5);
fuzzy->addFuzzyRule(fuzzyRule05);
//8. IF error=error5 and Derror=derror4 THEN velocity=slow
FuzzyRuleAntecedent* iferrorerror5AndDerrorderror4 = new
FuzzyRuleAntecedent();
iferrorerror5AndDerrorderror4->joinWithAND(error5, derror4);
FuzzyRuleConsequent* thenVelocitySlow6 = new
FuzzyRuleConsequent();
thenVelocitySlow6->addOutput(sedang);
// Instantiating a FuzzyRule object

```



```

FuzzyRule* fuzzyRule06 = new FuzzyRule(6,
iferrorerror5AndDerrorderror4, thenVelocitySlow6);
fuzzy->addFuzzyRule(fuzzyRule06
//11. IF error=error3 and Derror=derror5 THEN velocity=slow
FuzzyRuleAntecedent* iferrorerror3AndDerrorderror5 = new
FuzzyRuleAntecedent();
iferrorerror3AndDerrorderror5->joinWithAND(error3, derror5);
FuzzyRuleConsequent* thenVelocitySlow7 = new
FuzzyRuleConsequent();
thenVelocitySlow7->addOutput(no);
// Instantiating a FuzzyRule object
FuzzyRule* fuzzyRule07 = new FuzzyRule(7,
iferrorerror3AndDerrorderror5, thenVelocitySlow7);
fuzzy->addFuzzyRule(fuzzyRule07)
//12. IF error=error4 and Derror=derror5 THEN velocity=slow
FuzzyRuleAntecedent* iferrorerror4AndDerrorderror5 = new
FuzzyRuleAntecedent();
iferrorerror4AndDerrorderror5->joinWithAND(error4, derror5);
FuzzyRuleConsequent* thenVelocitySlow8 = new
FuzzyRuleConsequent();
thenVelocitySlow8->addOutput(sedang);
// Instantiating a FuzzyRule object
FuzzyRule* fuzzyRule08 = new FuzzyRule(8,
iferrorerror4AndDerrorderror5, thenVelocitySlow8);
fuzzy->addFuzzyRule(fuzzyRule08);

//13. IF error=error5 and Derror=derror5 THEN velocity=netral
FuzzyRuleAntecedent* iferrorerror5AndDerrorderror5 = new
FuzzyRuleAntecedent();
iferrorerror5AndDerrorderror5->joinWithAND(error5, derror5);
FuzzyRuleConsequent* thenVelocityNormal9 = new
FuzzyRuleConsequent();
thenVelocityNormal9->addOutput(sedang);
// Instantiating a FuzzyRule object
FuzzyRule* fuzzyRule09 = new FuzzyRule(9,
iferrorerror5AndDerrorderror5, thenVelocityNormal9);
fuzzy->addFuzzyRule(fuzzyRule09);
}

```

```

void display_sore()
{
  /*Controlled with fuzzy logic controller and gas sensors*/
  last_error = error;
  error = setpoint1 - ave;
  derror = error - last_error;

  Serial.println(error);
  Serial.println(derror);

  fuzzy->setInput(1, error);
  fuzzy->setInput(2, derror);
  fuzzy->fuzzify();
  //Tampilkan data Error
  Serial.print("Error: ");
  Serial.print(error3->getPertinence());   Serial.print(", ");
  Serial.print(error4->getPertinence());   Serial.print(", ");
  Serial.println(error5->getPertinence());   Serial.print(", ");
  //Tampilkan data Derror
  Serial.print("Derror: ");
  Serial.print(derror3->getPertinence());Serial.print(", ");
  Serial.print(derror4->getPertinence());   Serial.print(", ");
  Serial.println(derror5->getPertinence());   Serial.print(", ");

  output1 = fuzzy->defuzzify(1);
  Serial.print("KELUARAN AVERAGE: ");   Serial.println(output1);
  if(output1>=+4){
    blue.write("G");
    delay(8000);
  }
  else if(output1>2 && output<+4){
    blue.write("H");
    delay(4000);
  }
  else if(output1<=2){
    blue.write("T");
    delay(2000);
  }
}

```

LAMPIRAN B

Program Aktuator

```
#include <SoftwareSerial.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x3f, 16, 2);
SoftwareSerial blue(2,3);
char data;
int Relay = 4;

void setup() {
  Serial.begin(38400);
  blue.begin(38400);
  lcd.begin();
  pinMode(Relay, OUTPUT);
  digitalWrite(Relay, HIGH);
  lcd.setCursor(0,0);
  lcd.print("Sedang Menunggu");
  lcd.setCursor(0,1);
  lcd.print("Koneksi....");
}

void loop() {
  if(blue.available()>0)
  {
    //data = blue.read();
    //Serial.print(data);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Koneksi");
    lcd.setCursor(0,1);
    lcd.print("Tersambung !!");
    data = blue.read();
    Serial.print(data);

    if(data == 'A'){
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("Pagi & Kering");
    }
  }
}
```

```

lcd.setCursor(0,1);
lcd.print("Menyiram...");
digitalWrite(Relay, LOW);
delay(8000);
digitalWrite(Relay, HIGH);
}
else if(data == 'B'){
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Pagi & Agak Basah");
lcd.setCursor(0,1);
lcd.print("Menyiram...");
digitalWrite(Relay, LOW);
delay(4000);
digitalWrite(Relay, HIGH);
}
else if(data == 'C'){
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Pagi & Basah");
lcd.setCursor(0,1);
lcd.print("Mati...");
digitalWrite(Relay, HIGH);
delay(2000);
}
else if(data == 'D'){
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Siang & Kering");
lcd.setCursor(0,1);
lcd.print("Menyiram...");
digitalWrite(Relay, LOW);
delay(8000);
digitalWrite(Relay, HIGH);
}
else if(data == 'E'){
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Siang & Agak Basah");

```

```

lcd.setCursor(0,1);
lcd.print("Menyiram...");
digitalWrite(Relay, LOW);
delay(4000);
digitalWrite(Relay, HIGH);
}
else if(data == 'F'){
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Siang & Basah");
lcd.setCursor(0,1);
lcd.print("Mati...");
digitalWrite(Relay, HIGH);
delay(2000);
}
else if(data == 'G'){
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Sore & Kering");
lcd.setCursor(0,1);
lcd.print("Menyiram...");
digitalWrite(Relay, LOW);
delay(8000);
digitalWrite(Relay, HIGH);
}
}
else if(data == 'H'){
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Sore & Agak Basah");
lcd.setCursor(0,1);
lcd.print("Menyiram...");
digitalWrite(Relay, LOW);
delay(4000);
digitalWrite(Relay, HIGH);
}
}
else if(data == 'I'){
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Sore & Basah");

```

```
    lcd.setCursor(0,1);  
    lcd.print("Mati...");  
    digitalWrite(Relay, HIGH);  
    delay(2000);  
  }  
}  
}
```

LAMPIRAN C

Program Processing

```
import processing.serial.*;

Serial uart;
int theKey;
String theBuf;
int theSize = 8;
int theScale = 80;
float[][] theBitmap = new float[8][8];
float theTemp = 27.0;
float theAdj = 75;
float min = 0xffff;
float max = 0;
float ave = 0;
int max_x;
int max_y;

int state = 0;

void setup() {
  size(640, 640);
  textSize(24);

  print(Serial.list());
  String uartName = Serial.list()[0]; // <-- may have to modify this
  depending on the com port of the board
  print("\n\rSelected:" + uartName);
  uart = new Serial(this, uartName, 38400);
  uart.bufferUntil('\n');
}

void draw() {

  min = 0xffff;
  max = 0;

  ave = 0;
  for (int i=0; i<theSize; i++) {
```

```

for (int j=0; j<theSize; j++) {
    ave += theBitmap[i][j];
    if (theBitmap[i][j] < min)
    {
        min = theBitmap[i][j];
    }
    if (theBitmap[i][j] > max)
    {
        max = theBitmap[i][j];
        max_x = i;
        max_y = j;
    }
}
}
ave /= (8*8);

```

```

background(0);
for (int i=0; i<theSize; i++) {
    for (int j=0; j<theSize; j++) {
        float temp = theBitmap[i][j];
        float r = 0;
        float b = 0;
        float g = 0;

        g = map(temp, min, max, 0, 75);

        if (temp > ave) {
            r = map(temp, min, max, 0, 255);
        } else if (temp < ave) {
            b = map(temp, min, max, 0, 255);
        }

        stroke(r, g, b);
        fill(r, g, b);
        rect(i*theScale, (theSize - 1 - j)*theScale, theScale, theScale);
        stroke(255);
        fill(255);

        // uncomment this line for celcius display

```



```

        //text(int(temp), (i+0.1)*theScale,(j+0.5)*theScale);
        text(int(temp), (i+0.1)*theScale, ((theSize - 1
        j)+0.5)*theScale);
        // uncomment this line for fahrenheit display
        // text(int(1.8*temp+32),
        (i+0.1)*theScale,(j+0.85)*theScale);

        if((i == max_x) && (j==max_y))
        {
            stroke(255, 255, 0);
            fill(255, 255, 0);
            rect(i*theScale, (theSize - 1 - j)*theScale, theScale, theScale);
            stroke(0);
            fill(0);
            text(int(temp), (i+0.1)*theScale, ((theSize - 1
            j)+0.5)*theScale);
        }
    }
}

void serialEvent(Serial port) {
    theBuf = trim(port.readString());
    println(theBuf);
    if (theBuf.length() == 9) {
        if (theBuf.equals("Grid-EYE: ")) state = 1;
    } else if (state > 0 && state <= 8) {
        float[] pix = float(split(theBuf, ' '));
        for (int i=0; i<theSize; i++) {
            theBitmap[i][state-1] = pix[i];
            // theBitmap[i][state-1] = (pix[i]-20)*16;
        }
        state++;
    } else if (state == 9) {
        // get ambient temp
        // theTemp = float(theBuf);
        // println(pix[i]);
        state = 0;
    }
}
}

```

```
void keyPressed() {  
    uart.write(key);  
    theKey = key;  
}
```

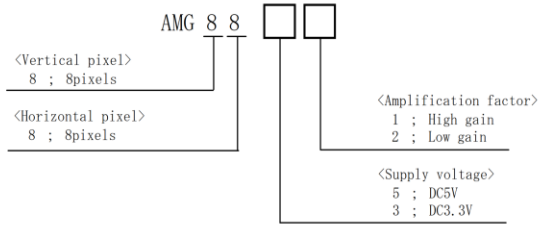
LAMPIRAN D

Datasheet AMG8833

1. Name : Infrared Array Sensor "Grid-EYE"

2. Part No. :

2-1 Part No. System



2-2 Part No. List

AMG8831 (VDD=3.3V , High gain type)

AMG8832 (VDD=3.3V , Low gain type)

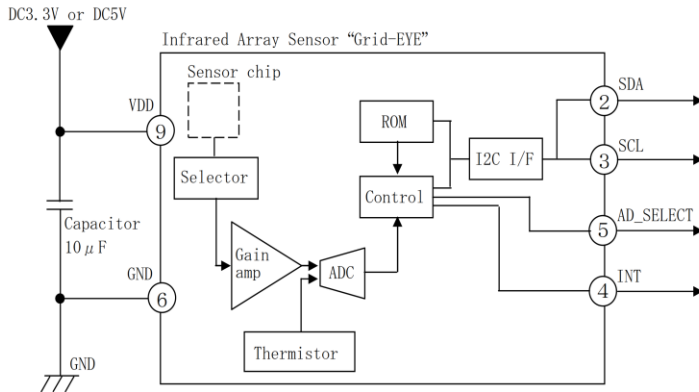
AMG8851 (VDD=5V , High gain type)

AMG8852 (VDD=5V , Low gain type)

3. Drawing : AMG8851 Product drawing

4. Characteristics

4-1 Internal circuit



*④INT terminal normally has same voltage as VDD. When interrupting, same as GND(0V).

*Regarding of recommended external circuit, please refer to section 4-9.

4-2 Main Functions

Item	Value
Pixel number	64 (8×8 Matrix)
External Interface	I ² C (fast mode)
Frame rate	Typ. 10 frames/sec or Typ. 1 frame/sec
Operating Mode	Normal Sleep Stand-by (10sec or 60sec intermittence)
Output Mode	Temperature Output
Calculate Mode	No moving average or Twice moving average
Temperature Output Resolution	0.25°C
Number of Sensor Addresses	2 (I ² C Slave Address)
Thermistor Output Temperature Range	-20°C~80°C
Thermistor Output Resolution	0.0625°C

4-3 Absolute Maximum Ratings

Item	Specification	Terminal
Applied voltage	-0.3~6.5V	VDD, VPP
Input/Output voltage	-0.3~Vdd+0.3V	SCL, SDA, AD_SELECT
Output current	-10~10mA	INT, SDA
ESD (Human Body Model)	1kV	All Terminals
ESD (Machine Model)	200V	All Terminals

4-4 Ratings

Item	Specification	
	High gain	Low gain
Applied voltage	3.3V±0.3V or 5.0V±0.5V	
Temperature Range of Measuring Object	0°C~80°C	-20°C~100°C
Operating temperature	0°C~80°C	-20°C~80°C
Storage temperature	-20°C~80°C	

4-5 Characteristics

Item	Specification	
	High gain	Low gain
Temperature Accuracy	Within Typ. $\pm 2.5^{\circ}\text{C}$	Within Typ. $\pm 3.0^{\circ}\text{C}$
Rated detection distance *1	5m (Max.)	
Field of View	Typ. 60° (Horizontal, Vertical)	
Optical Axis Gap	Within Typ. $\pm 5.6^{\circ}$ (Horizontal, Vertical)	
Current Consumption	Typ. 4.5mA (normal mode) Typ. 0.2mA (sleep mode) Typ. 0.8mA (stand-by mode)	
Setup Time	Typ. 50msec (Time to enable Communication after Setup) Typ. 15sec (Time to stabilize Output after Setup)	

- *1
- To have more than 4°C of temperature difference from background
 - Detection object size : $700 \times 250\text{mm}$ (Assumable human body size)

4-6 Electric characteristics

(1) Characteristics of the SDA and SCL I/O stages

parameter	symbol	Min.	Max.	unit
Low level input voltage	V_{IL}	-0.3	$0.3 \times V_{DD}$	V
High level input voltage	V_{IH}	$0.7 \times V_{DD}$	$V_{DD} + 0.3$	V
Hysteresis (SDA, SCL)	V_{hys}	$0.05 \times V_{DD}$		V
Low level output voltage (at 3mA sink current)	V_{OL}	0	0.4	V
Output fall time from V_{IHmin} to V_{ILmax} with a bus capacitance from 10pF to 400pF	t_{of}	$20 + C_b$	250	ns
Pulse width of spikes which must be suppressed by the input filter	t_{sp}	0	50	ns
Input current each I/O pin with an input voltage between $0.1 \times V_{DD} \sim 0.9 \times V_{DD}$	I_1	-10	10	μA
Capacitance for each I/O pin	C_i	-	10	pF

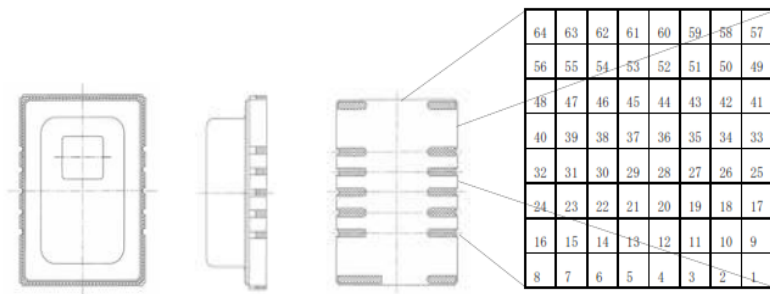
(2) Characteristics of the SDA and SCL bus lines

parameter	symbol	Min.	Max.	unit
SCL clock frequency	f_{SCL}	0	400	kHz
Hold time (repeated) START condition.	$t_{HD:STA}$	600	-	ns
Low period of the SCL clock	t_{LOW}	1.3		μs
High period of the SCL clock	T_{HIGH}	0.6		μs
Set-up time for a repeated START condition	$t_{SU:STA}$	0.6		μs
Data hold time	$t_{HD:DAT}$	0	900	ns
Data set-up time	$t_{SU:DAT}$	100		ns
Rise time of both SDA and SCL signals ($f_{SCL} > 100\text{kHz}$)	t_r	$20 + 0.1 \times C_b$	300	ns
Rise time of both SDA and SCL signals ($f_{SCL} \leq 100\text{kHz}$)	t_r		1000	ns
Fall time of both SDA and SCL signals	t_f	$20 + 0.1 \times C_b$	300	ns
Set-up time for STOP condition	$t_{SU:STO}$	600		ns
Bus free time between a STOP and START condition	t_{BUF}	1300		ns
Capacitive load for each bus line	C_b		400	pF

4-7 Pixel Array & Viewing Field

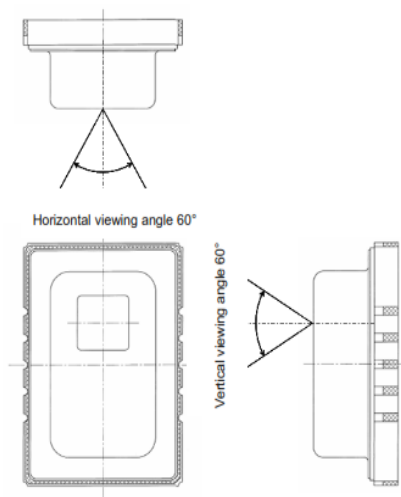
(1) Pixel Array

Pixel Array from 1 to 64 is shown below.



(2) Viewing Field

Sensor Viewing Field (Typical) is shown below.

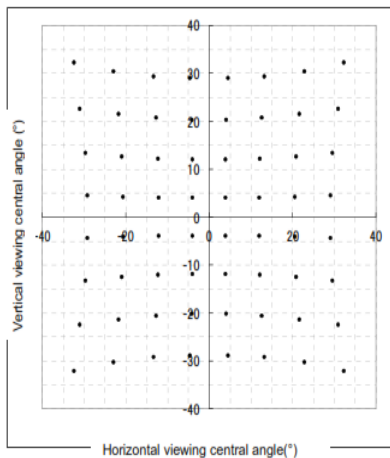


(3) Typical characteristics : Each pixel's viewing central angle

*Regarding of Pixel Array, please refer to 4-7(1).

Sensor's optical center (the origin of graph below) gap

: within Typ. $\pm 5.6^\circ$ (Both of horizontal and vertical directions)

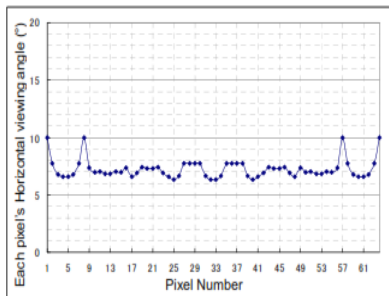


(4) Typical characteristics : Each pixel's viewing angle (half angle)

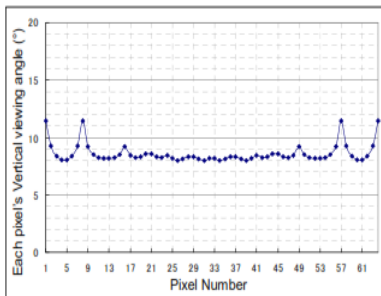
Central 4 pixels (Pixel No. 28, 29, 36, 37) viewing angle (half angle) :

horizontal direction Typ. 7.7°

vertical direction Typ. 8°



Each pixel's horizontal viewing angle



Each pixel's vertical viewing angle

Panasonic Infrared Array Sensor Grid-EYE (AMG88)

Infrared Array Sensor Grid-EYE



High Precision Infrared Array Sensor based on Advanced MEMS Technology

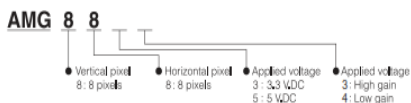
Features

- Temperature detection of two-dimensional area: 8 × 8 (64 pixels)
- Digital output (capability of temperature value output)
- Compact SMD package (adaptively to reflow mounting)
- RoHS compliant

Typical Applications

- High function home appliances (microwaves and air-conditioners)
- Energy saving at office (air-conditioning/lighting control)
- Digital signage
- Automatic doors/elevators

Ordering Information



Types

Product name	Number of pixel	Operating voltage	Amplification factor	Part number
Infrared array sensor Grid-EYE	64 (Vertical 8 × Horizontal 8 Matrix)	3,3 V,DC	High gain	AMG8833
			Low gain	AMG8834
		5,0 V,DC	High gain	AMG8853
			Low gain	AMG8854

Tape and reel package : 1,000 pcs.

Rating

Item	Performance	
	High gain	Low gain
Applied voltage	3,3 V,DC±0,3 V,DC or 5,0 V,DC±0,5 V,DC	
Temperature range of measuring object	0 °C to 80 °C +32 °F to +176 °F	-20 °C to 100 °C -4 °F to +212 °F
Operating temperature range	0 °C to 80 °C +32 °F to +176 °F	-20 °C to 80 °C -4 °F to +176 °F
Storage temperature range	-20 °C to 80 °C -4 °F to +176 °F	-20 °C to 80 °C -4 °F to +176 °F

Panasonic Infrared Array Sensor Grid-EYE (AMG88)

Absolute Maximum Ratings

Item	Absolute maximum ratings	Terminal
Applied voltage	-0,3 V _{DC} to 6,5 V _{DC}	VDD
Input voltage	-0.3 V _{DC} to VDD +0.3 V _{DC}	SCL, SDA, AD_SELECT
Output sink current	-10 mA to 10 mA	INT, SDA
Static electricity (Human body model)	1 kV	All terminals
Static electricity (Machine model)	200 V	All terminals

Characteristics

Item	Performance	
	High gain	Low gain
Temperature accuracy	Typical ± 2.5 °C ± 4.5 °F	Typical ± 3.0 °C ± 5.4 °F
Human detection distance *1	Max. 7 m 22.966 ft	
Viewing angle	Typical 60 °	
Optical axis gap	Within Typical ± 5.6 °	
Current consumption	Typical 4,5 mA (normal mode) Typical 0,2 mA (sleep mode) Typical 0,8 mA (stand-by mode)	
Setup time	Typical 50 ms (Time to enable communication after setup) Typical 15 s (Time to stabilize output after setup)	

Note: *1 To have more than 4 °C 7,2 °F of temperature difference from background
 Detection object size: 700 × 250 mm 27,559 × 9,843 inch (Assumes human body size)

Performance

Item	Performance
Number of pixel	64 (Vertical 8 × Horizontal 8 Matrix)
External interface	I ² C (fast mode)
Frame rate	Typical 10 frames/sec or 1 frame/sec
Operating mode *1	Normal Sleep Stand-by (10 sec or 60 sec intermittence)
Output mode	Temperature output
Calculate mode	No moving average or Twice moving average
Temperature output resolution	0,25 °C
Number of sensor address	2 (I ² C slave address)
Thermistor output temperature range	-20 °C to 80 °C -4 °F to +176 °F
Thermistor output resolution	0,0625 °C

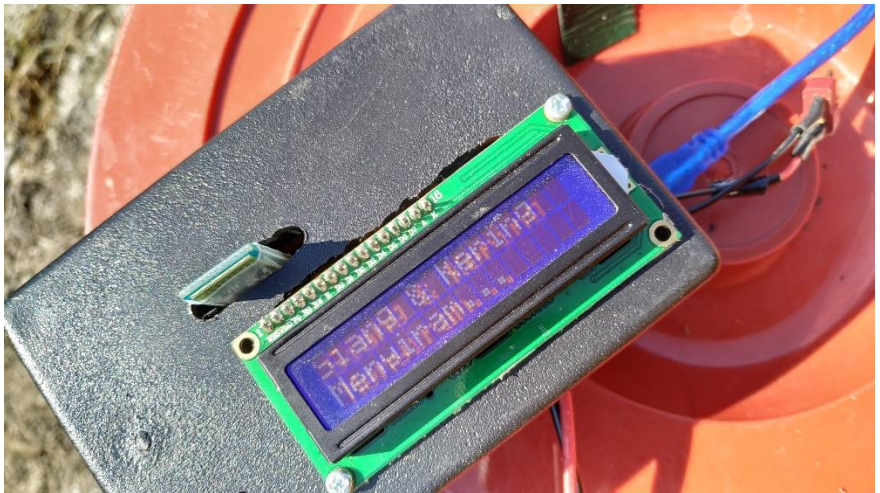
Note: *1 Normal Mode: normal operation mode; Sleep Mode: detection is off (output and data reading not possible); Standby Mode: 1 frame measuring intermittently every 10 or 60 sec.

LAMPIRAN E
Dokumentasi Tugas Akhir





























BIODATA PENULIS



Ricky Julio Rajagukguk lahir di Parindu pada tanggal 3 Juli 1997 merupakan anak kedua dari 3 bersaudara. Penulis menyelesaikan pendidikan dasar di SDN 69 Emplasmant, dilanjutkan pendidikan tingkat menengah di SMPK Makedonia , dan sekolah tingkat atas di SMAK Makedonia. Penulis memulai kehidupan perkuliahan pada tahun 2015 di Departemen Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember Surabaya. Selama masa perkuliahan, penulis aktif dalam berbagai kegiatan kepanitiaan dan organisasi baik di kampus maupun di luar kampus. Penulis juga pernah mengikuti lomba Internasional bersama tim Paduan Suara Mahasiswa ITS dan mendapat juara di tiga kategori.

Email : rickyrjagukguk@gmail.com
Hp/WA : 082230676210
Facebook : Ricky Rajagukguk