

***PROTOTYPE* SISTEM PENGATURAN SUHU PADA *AIR CONDITIONER*
SECARA OTOMATIS MENGGUNAKAN PENERAPAN METODE
KONTROL LOGIKA *FUZZY* BERDASARKAN KONDISI SUHU UDARA
LUAR, SUHU UDARA DIDALAM RUANGAN DAN JUMLAH PENGHUNI
RUANGAN BERBASIS ARDUINO UNO R3**



Disusun Oleh :

Reza Fahlevi

5115134301

Skripsi Ini Ditulis Untuk Memenuhi Sebagian Persyaratan

Dalam Mendapatkan Gelar Sarjana

PROGRAM STUDI PENDIDIKAN TEKNIK ELEKTRO

FAKULTAS TEKNIK

UNIVERSITAS NEGERI JAKARTA



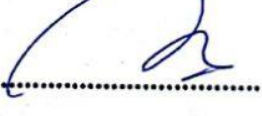

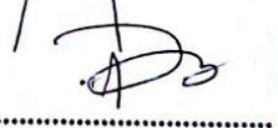
2017

HALAMAN PENGESAHAN

**PROTOTYPE SISTEM PENGATURAN SUHU PADA AIR CONDITIONER
SECARA OTOMATIS MENGGUNAKAN PENERAPAN METODE
KONTROL LOGIKA FUZZY BERDASARKAN KONDISI SUHU UDARA
LUAR, SUHU UDARA DIDALAM RUANGAN DAN JUMLAH PENGHUNI
RUANGAN BERBASIS ARDUINO UNO R3**

REZA FAHLEVI / 5115134301

PANITIA UJIAN SKRIPSI

NAMA DOSEN	TANDA TANGAN	TANGGAL
Drs. Purwanto G., M.T. (Ketua Penguji)		25.08.2017
Massus Subekti, M.T. (Sekretaris)		28-08-2017
Aris Sunawar, M.T. (Dosen Ahli)		29-08-2017
Muhammad Djaohar, S.T., M.Sc. (Dosen Pembimbing I)		30-08-2017
Dr. Muhammad Rif'an, S.T., M.T. (Dosen Pembimbing II)		28-08-2017

Tanggal Lulus :

HALAMAN PERNYATAAN

Dengan ini saya menyatakan bahwa :

1. Karya tulis skripsi / karya inovatif yang saya buat adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik sarjana, baik di Universitas Negeri Jakarta ataupun di perguruan tinggi lain.
2. Karya tulis ini adalah murni gagasan, rumusan dan penelitian saya sendiri dengan arahan dari dosen pembimbing.
3. Dalam karya tulis ini tidak terdapat karya atau pendapat yang telah di tulis atau dipublikasikan orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka.
4. Pernyataan ini saya buat dengan sesungguhnya dan apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah di peroleh karena karya tulis ini, serta sanksi lainnya sesuai dengan norma yang berlaku di Universitas Negeri Jakarta.

Jakarta, 1 Agustus 2017

Yang membuat pernyataan



Reza Fahlevi

5115134301

KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Allah SWT yang telah melimpahkan rahmat, hidayah serta karunia-Nya, sehingga penulis dapat menyelesaikan skripsi ini. Sholawat dan salam semoga selalu tercurahkan kepada nabi Muhammad SAW.

Penulisan skripsi ini bertujuan sebagai persyaratan kelulusan untuk menyelesaikan studi S1 Program Studi Pendidikan Teknik Elektro, Fakultas Teknik, Universitas Negeri Jakarta dan juga memberikan kesempatan kepada mahasiswa agar dapat mengimplementasikan berbagai macam ilmu yang telah didapatkan saat dibangku kuliah serta menambah wawasan dan dapat memaparkan hasil dari pengetahuan yang didapatkan selama melakukan penelitian dalam bentuk sebuah skripsi.

Pada kesempatan kali ini, penulis ingin mengucapkan banyak terima kasih atas bantuan diberikan kepada penulis, terutama kepada :

1. Allah SWT, yang selalu memberikan rahmat, hidayah serta karunia-Nya
2. Kedua orang tua yang terus memberikan dukungan baik moral ataupun material.
3. Bapak Muhammad Djaohar, S.T, M.Sc. selaku Dosen Pembimbing I dan terus memberikan dukungan, masukan dan arahan yang sangat bermanfaat
4. Bapak Dr. Muhammad Rif'an, S.T, M.T. selaku Dosen Pembimbing II dan terus memberikan dukungan, masukan dan arahan yang sangat bermanfaat
5. Teman-teman program studi Pendidikan Teknik Elektro 2013 yang selalu memberikan motivasi dan semangat
6. Teman-teman tim robotika Universitas Negeri Jakarta yang selalu memberikan semangat positif dan memberikan sangat banyak ilmu yang bermanfaat
7. Seluruh pihak lain yang telah ikut membantu penulis dalam menyelesaikan penelitian serta penyusunan skripsi ini, yang tidak dapat saya sebutkan satu-persatu

Penulis menyadari masih sangat banyak kekurangan dalam penulisan skripsi ini, baik dalam sistematika penulisan maupun isi materinya. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang membangun dari pembaca untuk menyempurnakan skripsi ini dimasa yang akan datang.

Akhirnya, penulis berharap semoga penulisan skripsi ini dapat bermanfaat bagi semua pembaca dan penulis mendoakan agar semua pihak yang ikut terlibat dalam proses pembuatan skripsi ini mendapatkan balasan rahmat dari Allah SWT.

Jakarta, 1 Agustus 2017

Penulis

ABSTRAK

REZA FAHLEVI, *Prototype Sistem Pengaturan Suhu Air Conditioner Secara Otomatis Menggunakan Penerapan Metode Kontrol Logika Fuzzy Berdasarkan Kondisi Suhu Udara Luar, Suhu Udara Didalam Ruangan Dan Jumlah Penghuni Ruangan Berbasis Arduino Uno R3*, Skripsi. Jakarta: Fakultas Teknik, Universitas Negeri Jakarta, 2017. Dosen Pembimbing : Moch. Djaohar, S.T., M.Sc. dan Dr. Muhammad Rifan, S.T., M.T.

Air Conditioner (AC) merupakan sebuah alat yang digunakan sebagai pendingin ruangan untuk menunjang kenyamanan bagi pengguna ruangan. Pada umumnya, pengaturan suhu pada *Air Conditioner* (AC) dilakukan secara manual dan pengaturan suhu pada *Air Conditioner* (AC) dilakukan secara asal dan tidak melihat kondisi disekitar. Oleh karena itu, *Air Conditioner* (AC) yang seharusnya merupakan alat yang dapat menunjang kenyamanan akan menjadi alat yang dapat mengganggu kenyamanan dari pengguna ruangan tersebut. Untuk mendapatkan nilai suhu yang ideal pada *Air Conditioner* (AC), para pengguna ruangan harus mengetahui bagaimana kondisi suhu udara diluar ruangan tersebut dan melihat jumlah manusia yang ada pada ruangan tersebut. Metode yang digunakan dalam penelitian ini adalah metode rekayasa teknik.

Penelitian ini bertujuan untuk menghasilkan sebuah alat yang dapat menyesuaikan *setting* suhu pada *Air Conditioner* (AC) dengan memperhatikan tiga variabel masukan yaitu : kondisi suhu udara luar, kondisi suhu udara didalam ruangan dan jumlah manusia yang terdapat didalam ruangan dengan menggunakan penerapan metode kontrol Logika *Fuzzy*. Kelebihan dari metode kontrol Logika *Fuzzy* adalah metode ini dapat melakukan penyesuaian keluaran berdasarkan kemungkinan-kemungkinan yang akan diberikan oleh variabel masukannya sehingga Logika *Fuzzy* dianggap cocok untuk memecahkan suatu permasalahan seperti penyesuaian *setting* suhu pada *Air Conditioner* (AC). Untuk pembacaan nilai suhu digunakan sensor suhu dengan jenis *Thermocouple Type K* yang telah diuji tingkat akurasi pembacaannya sehingga pembacaan suhu menjadi lebih akurat.

Hasil dari penelitian ini adalah sistem mampu mengatur nilai suhu pada *Air Conditioner* (AC) dengan merespon perubahan yang terjadi dari tiga variabel masukan, yaitu : suhu udara luar, suhu udara didalam ruangan serta jumlah penghuni didalam ruangan.

Kata Kunci : *Prototype, Air Conditioner* (AC), Logika *Fuzzy*, Suhu

ABSTRACT

REZA FAHLEVI, *Prototype Of Automatic Air Conditioner Temperature Setting System With Fuzzy Logic Control Method Based Outside Temperature Conditions, Indoor Temperature Conditions And The People Quantity Using Arduino Uno R3*, Skripsi. Jakarta: Faculty of Engineering, State University of Jakarta, 2017. Supervisor: Moch. Djaohar S.T., M.Sc. and Dr. Muhammad Rif'an, S.T., M.T.

Air Conditioner (AC) is an item which used for room cooler to make the person inside the room feel comfort. Commonly, temperature of Air Conditioner (AC) is set randomly and do not see the environment. Therefore, Air Conditioner (AC) that should give comfort will be uncomfortable. To get an ideal temperature value, user should know how the outside conditions and how many people in the room. The research method is using method of engineering.

The purpose of this research is to produce an item which can adjust the temperature value of Air Conditioner based on conditions of outside the room and the people quantity in the room using Fuzzy Logic control method. Fuzzy Logic control method can adjust the output value based on the possibility given by the input value. This can be used to be a solution of Air Conditioner problem. For the temperature reading, the sensor used is Thermocouple type K that has been tested so the temperature reading becomes more accurate.

The result of this research is the system can adjust the temperature value of Air Conditioner by responding to the changes that occur from the three input variables, namely : outside air temperature, indoor air temperature and The People Quantity inside the room.

Keywords : Prototype, Air Conditioner (AC), Fuzzy Logic, Temperature

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN	ii
HALAMAN PERNYATAAN	iii
KATA PENGANTAR	iv
ABSTRAK	v
<i>ABSTRACT</i>	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR	xiii
DAFTAR LAMPIRAN.....	xv
BAB I <u>P</u> ENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Identifikasi Masalah.....	4
1.3 Pembatasan Masalah.....	5
1.4 Perumusan Masalah	5
1.5 Tujuan Penelitian	6
1.6 Manfaat Penelitian	6
BAB II <u>T</u> INJAUAN PUSTAKA.....	8
2.1 <i>Air Conditioner</i> (AC).....	8
2.1.1 Pengertian Dasar Tentang <i>Air Conditioner</i> (AC).....	8
2.1.2 Faktor yang Mempengaruhi Beban <i>Air Conditioner</i> (AC).....	9
2.1.3 Perhitungan Daya Kompresor Pada <i>Air Conditioner</i> (AC)	10

2.2	Suhu Udara	12
2.2.1	Faktor – Faktor Yang Mempengaruhi Suhu Udara	13
2.2.2	Alat Pengukur Suhu.....	14
2.3	Ruangan.....	16
2.4	Sistem	17
2.5	Sistem Kontrol.....	18
2.6	Logika <i>Fuzzy</i>	20
2.7	<i>Fuzzy Logic Controller</i>	23
2.8	Pemrograman MATLAB	24
2.9	<i>Fuzzy Logic</i> Menggunakan MATLAB	26
2.9.1	<i>FIS Editor</i>	27
2.9.2	<i>Membership Function Editor</i>	29
2.10	<i>Arduino Uno R3</i>	29
2.11	<i>Arduino IDE</i>	30
2.12	<i>Thermocouple</i>	31
2.13	<i>Push Button</i>	32
2.14	<i>HC-05 Bluetooth Module</i>	32
2.15	<i>LCD (Liquid Crystal Display) 16x2</i>	33
2.16	Penelitian Yang Relevan.....	34
2.17	Kerangka Berpikir	36
BAB III METODE PENELITIAN		39
3.1	Waktu dan Tempat Penelitian.....	39
3.2	Alat dan Bahan Penelitian	39

3.3	Diagram Alir Penelitian	41
3.3.1	Studi Literatur	42
3.3.2	Pembuatan <i>Flowchart</i> Sistem Kerja Alat	42
3.3.3	Penentuan Rancangan Alat	42
3.3.4	Pendataan dan Pembelian Komponen Dan Bahan Yang Dibutuhkan .	43
3.3.5	Pembuatan Alat.....	44
3.3.6	Pengujian alat.....	44
3.3.7	Pengambilan Data	45
3.3.8	Analisis Data.....	45
3.3.9	Pembuatan Kesimpulan Dan Saran	45
3.4	Teknik dan Prosedur Pengumpulan Data	45
3.4.1	Diagram Blok Alat.....	45
3.4.2	Perancangan Perangkat Keras (<i>Hardware</i>)	47
3.4.3	Perancangan Perangkat Lunak (<i>Software</i>).....	50
3.4.4	Pengujian Alat dan Pengambilan Data	54
3.5	Teknik Analisis Data	63
BAB IV HASIL PENELITIAN		65
4.1	Deskripsi Hasil Penelitian.....	65
4.1.1	Hasil Pengujian Pengaruh Kondisi Suhu Udara Luar terhadap Nilai Suhu Udara Didalam Ruangan	65
4.1.2	Hasil Pengujian Pengaruh Jumlah Penghuni Ruangan terhadap Nilai Suhu Udara Didalam Ruangan	67
4.1.3	Hasil Pengujian Sensor <i>Thermocouple</i> 1 untuk pendeteksi suhu udara luar	68

4.1.4 Hasil Pengujian Sensor <i>Thermocouple</i> 2 untuk pendeteksi suhu udara didalam ruangan.....	70
4.1.5 Hasil Pengujian <i>Input Push Button</i>	71
4.1.6 Hasil Pengujian jarak pengiriman dan penerimaan data pada modul <i>bluetooth</i> HC-05	73
4.1.7 Hasil Pengujian <i>Input</i> Suhu dan Jumlah Penghuni Ruang Terhadap Nilai <i>setting</i> AC	74
4.1.8 Cara Kerja Alat	74
4.2 Analisis Data Penelitian.....	75
4.2.1 Analisis Sensor Suhu	75
4.2.2 Analisis <i>Push Button</i>	75
4.2.4 Analisis Perubahan Nilai <i>Input</i> Terhadap Nilai <i>Output</i>	76
4.3 Pembahasan	77
4.4 Aplikasi Hasil Penelitian	78
BAB V <u>KESIMPULAN DAN SARAN</u>	79
5.1 Kesimpulan	79
5.2 Saran	79
DAFTAR PUSTAKA	80
LAMPIRAN-LAMPIRAN	81

DAFTAR TABEL

Tabel 2.1	Fungsi dan Konfigurasi Pin-Pin <i>LCD</i>	34
Tabel 3.1	Daftar Komponen Yang Dibutuhkan	43
Tabel 3.2	Daftar Pin I/O Arduino Dengan <i>Bluetooth Transmitter</i> yang Diintegrasikan Dengan <i>Input Dan Output</i>	49
Tabel 3.3	Daftar Pin I/O Arduino Dengan <i>Bluetooth Receiver</i> Yang Diintegrasikan Dengan <i>Input Dan Output</i>	49
Tabel 3.4	Hasil pengujian kondisi suhu udara luar terhadap suhu udara didalam ruangan	57
Tabel 3.5	Data hasil pengamatan pengujian jumlah penghuni ruangan terhadap suhu udara didalam ruangan	58
Tabel 3.6	Data hasil pengamatan pengujian nilai tegangan dan pembacaan sensor suhu luar	59
Tabel 3.7	Data hasil pengamatan pengujian nilai tegangan dan pembacaan sensor suhu didalam ruangan	60
Tabel 3.8	Data hasil pengamatan pengujian <i>input push button (pull down resistor)</i>	61
Tabel 3.9	Data hasil pengamatan pengujian <i>input push button (pull up resistor)</i> ...	61
Tabel 3.10	Data hasil pengujian jarak pengiriman dan penerimaan data pada modul <i>bluetooth HC-05</i>	62
Tabel 3.11	Data hasil pengamatan pengujian input suhu dan jumlah penghuni terhadap nilai <i>setting AC</i>	63
Tabel 4.1	Data hasil pengamatan pengaruh kondisi suhu udara luar terhadap nilai suhu didalam ruangan	65
Tabel 4.2	Data hasil pengamatan pengaruh jumlah penghuni ruangan terhadap nilai suhu didalam ruangan	67
Tabel 4.3	Data hasil pengujian pembacaan sensor <i>thermocouple 1</i>	69
Tabel 4.4	Data hasil pengamatan pengujian pembacaan sensor <i>thermocouple 2</i>	70
Tabel 4.5	Data hasil pengamatan pengujian <i>input push button (pull down resistor)</i>	72

Tabel 4.6	Data hasil pengamatan pengujian <i>input push button (pull up resistor)</i> ...	72
Tabel 4.7	Pengujian jarak pengiriman dan penerimaan data pada modul <i>bluetooth HC-05</i>	73
Tabel 4.8	Data hasil Pengujian Input Suhu dan Jumlah Penghuni Ruangan Terhadap Nilai <i>Setting AC</i>	74

DAFTAR GAMBAR

Gambar 2.1 <i>Thermometer Air Raksa</i>	15
Gambar 2.2 <i>Thermistor</i>	16
Gambar 2.3 <i>Resistive Temperature Detector</i>	16
Gambar 2.4 Sistem Kontrol Terbuka (<i>Open Loop</i>)	19
Gambar 2.5 Sistem Kontrol Tertutup (<i>Closed Loop</i>).....	19
Gambar 2.6 Diagram Blok Logika <i>Fuzzy</i>	22
Gambar 2.7 Diagram <i>Fuzzy Logic Controller</i>	23
Gambar 2.8 Tampilan Awal MATLAB.....	24
Gambar 2.9 <i>Fis Editor</i>	27
Gambar 2.10 Diagram Fis AC	28
Gambar 2.11 <i>Arduino Uno R3</i>	30
Gambar 2.12 <i>Form Splash Arduino IDE 1.8.0</i>	31
Gambar 2.13 <i>Thermocouple</i>	31
Gambar 2.14 <i>Push Button</i>	32
Gambar 2. 15 <i>HC-05 Bluetooth Module</i>	33
Gambar 2.16 <i>LCD 16x2</i>	33
Gambar 2.17 Diagram Blok Kerangka Berpikir	38
Gambar 3.1 <i>Flowchart Penelitian</i>	41
Gambar 3.2 Desain <i>Box Remote Control Receiver</i>	42
Gambar 3.3 Desain <i>Box Remote Control Transmitter</i>	43
Gambar 3.4 Diagram Blok Alat	46
Gambar 3.5 <i>Rules (1 - 17)</i>	50
Gambar 3.6 <i>Rules (18 - 34)</i>	51
Gambar 3.7 <i>Rules (35 - 45)</i>	51
Gambar 3.8 <i>Surface</i>	51
Gambar 3.9 Fungsi Keanggotaan Suhu Luar	52
Gambar 3.10 Fungsi Keanggotaan Suhu Dalam.....	53
Gambar 3.11 Fungsi Keanggotaan Jumlah Penghuni	53
Gambar 3.12 Fungsi Keanggotaan <i>Setting AC</i>	54

Gambar 4.1	Grafik Pengaruh Suhu Luar Dengan Suhu Dalam	66
Gambar 4.2	Grafik Pengaruh Jumlah Penghuni Ruangan Terhadap Nilai Suhu Udara Dalam	68
Gambar 4.3	Grafik Perbandingan Antara <i>Thermocouple</i> 1 dan <i>Thermometer Digital</i>	69
Gambar 4.4	Grafik Perbandingan Antara <i>Thermocouple</i> 2 dan <i>Thermometer Digital</i>	71
Gambar 4.5	Skema <i>Push Button</i> Dengan <i>Pull Down Resistor</i>	72
Gambar 4.6	Skema <i>Push Button</i> Dengan <i>Pull Up Resistor</i>	73

DAFTAR LAMPIRAN

LAMPIRAN 1. Dokumentasi Produk Yang Dihasilkan.....	81
LAMPIRAN 2. Skematik Rangkaian	82
LAMPIRAN 3 Data-Data Pengukuran	83
LAMPIRAN 4. Data-Data Perhitungan.....	86
LAMPIRAN 5 Data Pendukung Lain Yang Berkaitan.....	87

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Dunia kita terbagi menjadi berbagai macam sudut pandang. Apa yang kita lihat dan kita rasakan belum tentu sama dengan apa yang dilihat dan dirasakan oleh orang lain. Contoh sederhananya adalah suhu. Suhu merupakan hal yang bersifat *fuzzy* atau samar karena suhu tidak dapat didefinisikan secara pasti.

Setiap orang akan mendefinisikan nilai suhu dengan sudut pandang yang berbeda. Misalnya, ketika kita bertanya kepada beberapa orang tentang pemikiran mereka terhadap suhu yang menunjukkan nilai 20°C , maka jawaban dari setiap orang akan berbeda. Orang pertama bisa saja menjawab bahwa suhu 20°C terdefinisi sebagai suhu yang dingin, orang kedua bisa menjawab bahwa suhu 20°C terdefinisi sebagai suhu yang sejuk dan orang ketiga bisa saja menjawab bahwa suhu 20°C terdefinisi sebagai suhu yang normal. Dari pendapat – pendapat tersebut kita tidak dapat memastikan definisi dari nilai 20°C pada suatu suhu karena suhu adalah hal yang bersifat *fuzzy* atau samar dan tidak bisa dipastikan.

Hal ini akan berpengaruh terhadap *setting* suhu disuatu ruangan yang menggunakan AC (*Air Conditioner*) ataupun jenis kipas pendingin yang lain dimana setiap orang akan merasakan hal yang berbeda terhadap suhu yang telah *diset*. Jika disuatu ruangan ada orang yang mendefinisikan suhu rendah (kisaran 16°C - 20°C) sebagai suhu yang normal untuk dirasakan, maka akan menjadi suatu hal yang mengganggu bagi orang – orang yang merasa suhu tersebut terlalu dingin bagi mereka

terutama mereka yang bisa jadi memiliki penyakit alergi terhadap udara dingin dan ditambah dengan adanya beberapa faktor yang dapat menyebabkan berubahnya nilai suhu pada ruangan tersebut, diantara adalah suhu udara luar dan jumlah penghuni ruangan.

Suhu udara yang berasal dari luar ruangan juga memiliki pengaruh yang besar terhadap kondisi suhu didalam ruangan. Suhu udara luar yang bernilai rendah akan berdampak terhadap kondisi suhu didalam ruangan yang juga akan bernilai rendah. Begitu juga pengaruh dari jumlah manusia yang menghuni ruangan tersebut. Jika penghuni didalam ruangan berjumlah sedikit, maka suhu didalam ruangan tersebut juga akan bernilai rendah karena jumlah kalor yang dihasilkan oleh penghuni sedikit dan semakin banyak jumlah manusia yang berada didalam ruangan tersebut, maka kalor yang akan dilepaskan oleh mereka akan semakin banyak sehingga suhu didalam ruangan tersebut akan meningkat akibat banyaknya jumlah kalor yang dihasilkan oleh para penghuni ruangan. Sehingga, pengaturan suhu dari sebuah AC (*Air Conditioner*) seharusnya dapat disesuaikan dengan kondisi yang ada.

Salah satu contoh kasus adalah ketika nilai suhu udara diluar ruangan rendah dan diruangan tersebut terdapat hanya sedikit orang, terkadang kita salah untuk menentukan setting AC atau kipas pendingin yang pas untuk kondisi tersebut. Kita dengan tidak sengaja menyetel AC atau kipas pendingin dengan suhu yang rendah padahal kondisi ruangan sepi dan suhu udara diluar ruangan yang juga rendah. Dan kembali lagi, permasalahannya akan timbul kepada mereka – mereka yang memiliki penyakit alergi terhadap suhu dingin. Ini membuat kita harus bisa berpikir untuk mencari solusi dari masalah – masalah tersebut.

Seiring berjalannya waktu, dengan perkembangan teknologi yang telah berkembang dengan sangat pesat dan dengan majunya pola pikir manusia sehingga kita dapat menciptakan berbagai macam ide dan pemikiran yang kreatif untuk membuat hal yang dapat menjadi solusi untuk setiap masalah didalam kehidupan ini. Dalam dunia elektro dan komputer, sistem kendali menjadi hal yang menjadi favorit bagi para *developer* atau pengembang teknologi. Dengan menggunakan sebuah sistem kendali dengan dukungan dari berbagai macam jenis metode kontrol membuat hal yang sulit untuk didefinisikan menjadi dapat terdefinisi.

Salah satu contoh metode yang sering digunakan untuk mendefinisikan sebuah hal yang bersifat samar adalah metode logika *fuzzy*. Metode logika *fuzzy* biasa diterapkan untuk menyelesaikan hal – hal yang mengandung unsur ketidakpastian, ketidaktepatan dan sebagainya. Metode ini juga logika *fuzzy* dianggap mendekati pola pikir manusia.

Logika *fuzzy* adalah suatu metode pemrograman untuk memetakan ruang input kedalam suatu ruang output (Fanani Arief G.,2015). Logika *fuzzy* merupakan sebuah logika yang berprinsip pada konsep kebenaran sebagian yang berbeda dengan logika *crisp* yang menyatakan bahwa segala sesuatu dapat diekspresikan sebagai suatu bilangan *crisp* (0 dan 1). Logika *fuzzy* merupakan jenis metode yang dapat digabungkan dengan beberapa metode lain seperti metode kontrol PID (*Proportional, Integrative, Derivative*).

Pada penelitian sebelumnya yang dilakukan oleh Maudi Mayangsari (2016) dengan judul penelitian *ANALISIS FUZZY LOGIC PADA PENGARUH SUHU UDARA LUAR DAN JUMLAH MANUSIA TERHADAP SUHU SETTING AIR*

CONDITIONER (AC) DI GEDUNG L1 JURUSAN TEKNIK ELEKTRO FAKULTAS TEKNIK UNIVERSITAS NEGERI JAKARTA telah didapatkan sebuah kesimpulan bahwa metode kontrol Logika *Fuzzy* (*Fuzzy Logic*) efektif digunakan sebagai sebuah metode kontrol yang diterapkan pada suatu mesin pendingin seperti *Air Conditioner* (AC).

Maka, berdasarkan pertimbangan tersebut logika *fuzzy* dipilih untuk menjadi suatu solusi untuk bisa mengatasi permasalahan perbedaan dalam pendefinisian suhu yang sering terjadi. Alat yang akan dibuat adalah *Prototype* Sistem Pengaturan Suhu *Air Conditioner* (AC) Secara Otomatis Menggunakan Penerapan Metode Kontrol Logika *Fuzzy* Berdasarkan Kondisi Suhu Udara Luar, Suhu Udara Didalam Ruangan Dan Jumlah Penghuni Ruangan Berbasis Arduino Uno R3.

1.2 Identifikasi Masalah

Berdasarkan latar belakang yang telah dikemukakan sebelumnya, ada beberapa permasalahan yang dapat diidentifikasi, yaitu:

1. Pengaruh suhu udara luar terhadap pengaturan tingkat pendinginan pada *AC* (*Air Conditioner*).
2. Pengaruh jumlah penghuni ruangan terhadap pengaturan tingkat pendinginan pada *AC* (*Air Conditioner*).
3. Cara membuat alat yang dapat mengatur suhu pada *AC* (*Air Conditioner*) secara otomatis berdasarkan nilai suhu udara luar, nilai suhu udara didalam ruangan serta jumlah penghuni ruangan.

1.3 Pembatasan Masalah

Berdasarkan latar belakang masalah dan identifikasi masalah diatas, peneliti membatasi permasalahan pada perancangan *prototype* sistem pengaturan suhu *Air Conditioner* (AC) otomatis berdasarkan kondisi suhu udara luar, suhu udara didalam ruangan dan jumlah orang dengan penerapan metode kontrol logika *fuzzy* berbasis *Arduino Uno R3*, yaitu :

1. Alat kendali yang digunakan dalam pembuatan rancangan *prototype* ini adalah *Arduino Uno R3* dan menggunakan sensor suhu dengan jenis *thermocouple type K* sebagai input suhu luar dan dalam ruangan.
2. Simulasi penghitungan jumlah penghuni ruangan dilakukan dengan menggunakan *push button*
3. Pembuatan logika program pada *Arduino Uno R3* tidak sepenuhnya disesuaikan dengan hasil perhitungan logika *fuzzy* melalui *software* MATLAB.
4. Menggunakan metode sugeno dalam proses inferensi logika *fuzzy*.
5. Sistem pengaturan suhu menggunakan *remote control* dari *Air Conditioner* (AC) yang telah dimodifikasi.

1.4 Perumusan Masalah

Berdasarkan latar belakang, identifikasi masalah dan pembatasan masalah yang telah dikemukakan, maka dapat dibuat sebuah rumusan masalah dalam penelitian ini adalah bagaimanakah cara membuat sebuah *prototype* sistem pengaturan suhu *Air Conditioner* (AC) otomatis menggunakan penerapan metode logika *fuzzy* berdasarkan

kondisi suhu udara luar, suhu udara didalam ruangan dan jumlah penghuni ruangan berbasis *Arduino Uno R3* ?

1.5 Tujuan Penelitian

Sesuai dengan permasalahan yang telah dirumuskan dan diidentifikasi, maka tujuan dari pembuatan tugas akhir ini adalah menghasilkan sebuah *prototype* alat yang dapat mengatur suhu pada *Air Conditioner* (AC) secara otomatis berdasarkan kondisi suhu udara luar, suhu udara didalam ruangan dan jumlah orang yang ada didalam ruangan tersebut.

1.6 Manfaat Penelitian

Penelitian ini bertujuan untuk membuat sebuah *prototype* alat yang dapat mengatur suhu *Air Conditioner* (AC) secara otomatis berdasarkan kondisi suhu udara luar, suhu udara didalam ruangan dan jumlah penghuni ruangan. Adapun manfaat dari penelitian ini adalah :

1. Untuk memperluas pengetahuan peneliti terutama mengenai sistem kendali dan metode-metode yang dapat diterapkan pada suatu sistem kendali, khususnya *Arduino Uno R3* dan Logika *Fuzzy*.
2. Untuk membuat sebuah *prototype* alat yang dapat mengatur suhu *Air Conditioner* (AC) secara otomatis berdasarkan kondisi suhu udara luar, suhu udara didalam ruangan dan jumlah penghuni ruangan.sebagai *prototype* awal yang nantinya dapat dikembangkan dan direalisasikan agar dapat menjadi suatu solusi yang tepat untuk permasalahan *setting* suhu pada *Air Conditioner* (AC) dan juga membuat salah satu contoh alat yang menggunakan penerapan metode kontrol Logika *Fuzzy* yang dapat

dijadikan sebagai referensi pembelajaran pada mata kuliah Logika *Fuzzy* di Rumpun Teknik Elektro Universitas Negeri Jakarta.

BAB II

TINJAUAN PUSTAKA

2.1 *Air Conditioner* (AC)

2.1.1 Pengertian Dasar Tentang *Air Conditioner* (AC)

Sistem Tata Udara (*Air Conditioning*) secara umum merupakan proses pengkondisian udara pada suatu ruangan melalui pengendalian suhu dan kelembapan udara sehingga mencapai kondisi dan nilai yang diinginkan. Alat yang digunakan untuk melakukan pengkondisian udara disebut sebagai *Air Conditioner* (AC). *Air Conditioner* (AC) dapat didefinisikan sebagai suatu rangkaian mesin yang memiliki fungsi sebagai pendingin udara yang berada di sekitar mesin pendingin tersebut.

Air Conditioner (AC) merupakan suatu mesin yang digunakan untuk mendinginkan udara dengan cara mensirkulasikan gas *refrigerant* yang berada di pipa yang ditekan dan dihisap oleh kompresor. Adapun sebab mengapa gas *refrigerant* dipilih sebagai bahan yang disirkulasikan, yaitu karena bahan ini mudah menguap dan bentuknya bisa berubah-ubah, yang berbentuk cairan dan gas. Panas yang berada pada pipa kondensor berasal dari gas *refrigerant* yang ditekan oleh kompresor sehingga bahan tersebut menjadi panas dan pada bagian *Automatic Expansion Valve* pipa tempat sirkulasi gas *refrigerant* diperkecil, sehingga tekanannya semakin meningkat dan pada pipa evaporator menjadi dingin.

Bagian-Bagian *Air Conditioner* (AC) Beserta Fungsinya :

a) *Compressor* (kompresi) yaitu berfungsi untuk memompa gas *refrigerant*.

- b) *Recervoir* yaitu berfungsi untuk menyimpan gas dari *condensor* sebelum dialirkan ke *compressor*.
- c) *Condensor* (penguapan) berfungsi untuk tempat pembuangan temperatur panas.
- d) *Evaporator* (pengembunan) berfungsi untuk tempat pembuangan temperatur dingin.
- e) *Filter Dryer* berfungsi sebagai penyaring sisa-sisa kotoran gas dan oli.
- f) *Motor Fan* dan *Blower* berfungsi untuk memutar kipas *fan* dan *blower* agar terjadinya sirkulasi udara.

Pada saat ini, teknologi pada *Air Conditioner (AC)* telah berkembang pesat. Salah satu diantaranya adalah adanya teknologi *inverter* pada *Air Conditioner (AC)* dimana secara otomatis *inverter* tersebut akan mengendalikan kecepatan dari motor yang berada didalam *Air Conditioner (AC)*. Dan dalam pengembangan selanjutnya dapat diterapkan berbagai metode kontrol yang bisa membuat teknologi *inverter* tersebut menjadi lebih baik lagi.

2.1.2 Faktor yang Mempengaruhi Beban *Air Conditioner (AC)*

Sebelum merencanakan atau memasang *Air Conditioner (AC)*, maka perlu mempertimbangkan beberapa hal berikut agar *Air Conditioner (AC)* tersebut bisa berfungsi maksimal dan efisien :

a. Ukuran Ruangan

Ukuran ruangan menentukan berapa banyak BTU (*british thermal unit*) atau kecepatan pendinginan. BTU adalah kecepatan pendinginan untuk ruangan satu meter persegi dengan tinggi *stuser* (umumnya tiga meter). Semakin besar suatu ruangan akan semakin besar pula nilai BTU yang dibutuhkan.

b. Beban Pendinginan

Beban pendinginan umumnya berasal dari dalam ruangan. Misalnya, jumlah manusia yang ada didalam ruangan tersebut dan jumlah penerangan didalam ruangan. Semakin banyak jumlah manusia didalam ruangan dan semakin banyak jumlah penerangan yang dinyalakan maka nilai kebutuhan BTU dari ruangan tersebut akan meningkat.

c. Posisi Ruangan

Posisi ruangan memiliki pengaruh yang cukup besar terhadap perhitungan nilai BTU dari ruangan tersebut. Ruangan yang terletak pada lantai 1 atau diapit oleh lantai atas akan lebih rendah koefisien kalornya dengan ruangan yang terletak pada lantai paling atas. Selain letak lantai posisi hadap ruangan juga mempengaruhi besar kecilnya koefisien kalor yang dihasilkan ruangan tersebut.

Matahari terbit dari Timur ini menyebabkan ruangan yang menghadap ke Barat akan mempunyai koefisien kalor paling besar ini karena sinaran matahari terbit akan mengarah ke arah Barat. Diikuti dengan ruangan yang menghadap ke Selatan. Lalu ruangan yang menghadap ke Timur akan lebih 16 kecil koefisien kalornya dari ruangan yang menghadap ke Barat dan Selatan, ini karena ruangan yang menghadap ke Timur hanya tersinar matahari pada saat mata hari tenggelam. Diikuti dengan ruangan yang menghadap ke Utara.

2.1.3 Perhitungan Daya Kompresor Pada *Air Conditioner* (AC)

Menurut Afendi (2012:13) yang disadur dari Maudi (2016) istilah PK atau HP atau daya kuda (*Paard Kracht/Daya Kuda/Horse Power* (HP) pada *Air Conditioner*

(AC) sebenarnya merupakan satuan daya pada kompresor *Air Conditioner (AC)* bukan daya pendingin *Air Conditioner (AC)*. Untuk daya pendingin *Air Conditioner (AC)* satuannya BTU/hr.

Rumus Kebutuhan BTU

$$\frac{LxWxIxHxE}{60} \quad (2.1)$$

Keterangan :

L = Panjang Ruangan (dalam feet, 1 meter = 3,28 feet)

W = Lebar Ruangan (dalam feet, 1 meter = 3,28 feet)

H = Tinggi Ruangan (dalam feet, 1 meter = 3,28 feet)

I = Nilai 10 jika ruangan berada di lantai bawah, atau berimpit dengan ruangan lain; Nilai 18 jika ruangan di lantai atas.

E = nilai 16 jika dinding terpanjang menghadap utara; nilai 17 jika dinding menghadap timur; nilai 18 jika menghadap selatan; nilai 20 jika menghadap barat. 60 = konstanta

Rumus kebutuhan BTU ruangan dengan beban manusia pada ruangan :

$$\text{Kebutuhan BTU Ruang Penuh} = (\text{Keb. BTU R.Kosong}) + (\text{Jumlah Manusia} \times \text{Kalor BTU pria}) + (\text{Jumlah Manusia} \times \text{Kalor BTU wanita}) \quad (2.2)$$

Rumus kebutuhan BTU ruangan dengan beban manusia dan beban elektronik pada ruangan :

$$\text{Kebutuhan BTU Total} = \text{Keb. BTU R. Penuh} + (\text{Jml lampu} \times \text{Kalor}) + (\text{Jml Elektronik} \times \text{Kalor}) + (\text{Jml Laptop} \times \text{Kalor}) \quad (2.3)$$

Kapasitas *Air Conditioner* (AC) berdasarkan PK adalah sebagai berikut :

$$\text{AC } \frac{1}{2} \text{ PK} = \pm 5.000 \text{ BTU/h}$$

$$\text{AC } \frac{3}{4} \text{ PK} = \pm 7.000 \text{ BTU/h}$$

$$\text{AC 1 PK} = \pm 9.000 \text{ BTU/h}$$

$$\text{AC } 1 \frac{1}{2} \text{ PK} = \pm 12.000 \text{ BTU/h}$$

$$\text{AC 2 PK} = \pm 18.000 \text{ BTU/h}$$

2.2 Suhu Udara

Temperatur atau suhu adalah suatu ukuran untuk tingkat panas suatu benda. Suhu suatu benda ialah keadaan yang menentukan kemampuan benda tersebut untuk mentransfer panas atau menerima panas, dari benda satu ke benda yang lain (A. Fadholi, 2013).

Temperatur atau suhu adalah ukuran energi kinetik rata – rata dari pergerakan molekul – molekul. Suhu suatu benda ialah keadaan yang menentukan kemampuan benda tersebut, untuk memindahkan atau transfer panas ke benda – benda lain atau menerima panas dari benda – benda lain tersebut. Dalam sistem dua benda, benda yang kehilangan panas dikatakan benda yang bersuhu lebih tinggi (M. Jefry & Diah R., 2013).

Dapat disimpulkan bahwa suhu merupakan suatu keadaan yang menentukan kemampuan suatu benda untuk untuk mentransfer atau menerima energi panas dari satu benda ke benda yang lain yang akan mempengaruhi tingkat panas benda tersebut. Jika benda tersebut mengalami kondisi kehilangan energi maka benda tersebut dapat dikatakan benda tersebut bersuhu tinggi karena benda tersebut mampu mentransfer energinya ke benda lain. Dan sebaliknya jika benda tersebut mengalami kondisi mendapatkan energi maka benda tersebut dapat dikatakan benda tersebut bersuhu rendah karena benda tersebut mampu menerima energi dari benda lain.

Satuan temperatur atau suhu dalam SI (Satuan Internasional) adalah *Kelvin* (K). Namun, masyarakat di Indonesia biasanya menggunakan derajat *Celcius* (C) sebagai satuan untuk suhu dan temperatur. Selain itu, satuan suhu lainnya adalah *Fahrenheit* (F) dan *Reamur* (R).

2.2.1 Faktor – Faktor Yang Mempengaruhi Suhu Udara

a) Sudut Datangnya Sinar Matahari

Sudut datang sinar matahari berada pada nilai terkecil saat pagi dan sore hari. Sedangkan nilai sudut datang sinar matahari terbesar terjadi pada siang hari. Pada siang hari, matahari berada pada titik tertinggi sehingga sinar yang dipancarkan ke bumi semakin maksimal sehingga akan menyebabkan terjadinya kenaikan suhu di bumi dan menyebabkan kondisi cuaca menjadi panas. Sedangkan pada pagi ataupun sore hari, sudut datang sinar matahari akan mengecil sehingga penyinaran yang terjadi kearah bumi akan semakin berkurang dan menyebabkan suhu yang diterima bumi akan semakin rendah.

b) Rendah Atau Tingginya Suatu Tempat

Suatu tempat yang berada pada kedudukan yang tinggi akan memiliki suhu udara yang lebih rendah dibandingkan tempat yang berada pada kedudukan yang rendah. Sebagai contoh adalah daerah pegunungan dan daerah pantai yang memiliki ketinggian yang berberbeda maka nilai suhu dikedua tempat itupun akan berbeda pula.

Perbedaan suhu udara yang disebabkan adanya perbedaan ketinggian pada suatu tempat disebut amplitudo. Garis khayal yang menghubungkan tempat-tempat yang memiliki nilai suhu udara yang sama disebut garis *isotherm*.

c) Awan

Awan merupakan sesuatu yang dapat menghalangi datangnya sinar matahari menuju bumi. Jika pada suatu tempat memiliki kondisi cuaca berawan, maka sinar matahari yang diterima tempat tersebut akan relatif sedikit, sehingga nilai suhu udara ditempat tersebut akan menjadi rendah.

d) Angin

Angin mempunyai pengaruh yang cukup besar terhadap perubahan suhu disuatu tempat. Ketika suatu tempat memiliki kondisi yang berangin maka tempat tersebut akan lebih dingin dibandingkan dengan tempat yang memiliki intensitas angin yang lebih rendah.

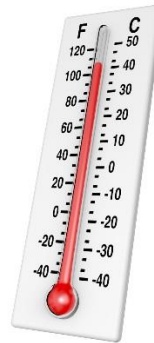
2.2.2 Alat Pengukur Suhu

Pengukuran suhu dapat dilakukan dengan alat pengukur suhu. Terdapat berbagai macam jenis alat pengukur suhu, diantaranya : *thermometer*, *thermostat* dan juga

terdapat beberapa jenis sensor yang dapat digunakan untuk mengukur suhu. Berikut ini adalah beberapa jenis alat ukur yang dapat digunakan untuk mengukur suhu :

1. *Thermometer*

Thermometer merupakan sebuah alat ukur yang digunakan untuk mengukur nilai suhu ataupun perubahan nilai suhu. *Thermometer* berasal dari bahasa Latin *thermo*, yang artinya panas dan *meter* yang artinya untuk mengukur. Jenis *thermometer* yang sering kita gunakan adalah jenis *thermometer* air raksa seperti pada gambar 2.1 berikut.



Gambar 2.1 *Thermometer* Air Raksa

(Sumber : <http://www.yms.co.za/product/fridge-thermometer/>)

2. *Thermistor (Thermal Resistor)*

Thermistor atau *Thermal Resistor* merupakan sebuah komponen elektronika yang nilai resistansinya dipengaruhi oleh kenaikan atau penurunan suhu. Thermistor pada dasarnya terdiri dari 2 jenis, yaitu :

- a. PTC (*Positive Temperature Coefficient*), sensor ini akan mengalami kenaikan nilai resistansi ketika suhu meningkat
- b. NTC (*Negative Temperature Coefficient*), sensor ini akan mengalami penurunan nilai resistansi ketika suhu meningkat

Contoh dari PTC dan NTC dapat dilihat pada gambar 2.2.



Gambar 2.2 *Thermistor*

(a) *Negative Temperature Coefficient*, (b) *Positive Temperature Coefficient*

(sumber : <http://teknikelektronika.com/pengertian-sensor-suhu-jenis-jenis-sensor-suhu/>)

3. **RTD (*Resistive Temperature Detector*)**

RTD (*Resistive Temperature Detector*) merupakan suatu sensor suhu yang memiliki prinsip kerja mirip dengan PTC, yaitu akan mengalami kenaikan nilai resistansi ketika terjadi peningkatan suhu. Letak perbedaan RTD dengan PTC adalah tingkat akurasi RTD yang lebih tinggi dibanding PTC. Contoh RTD dapat dilihat pada gambar 2.3.



Gambar 2.3 *Resistive Temperature Detector*

(sumber : <http://teknikelektronika.com/pengertian-sensor-suhu-jenis-jenis-sensor-suhu/>)

2.3 **Ruangan**

Menurut Prasetya (2009) yang diacu dalam Maudi (2016) ruang merupakan daerah 3 dimensi dimana obyek dan peristiwa berada. Ruang memiliki posisi serta arah yang relatif, terutama bila suatu bagian dari daerah tersebut dirancang sedemikian rupa untuk tujuan tertentu. Ruang merupakan wadah dari aktivitas-aktivitas manusia, baik aktivitas untuk kebutuhan fisik maupun emosi manusia. Ruang digunakan untuk mewadahi satu aktivitas manusia atau lebih. Ruang yang digunakan lebih dari satu

fungsi dan aktivitas disebut ruang multifungsi. Ruang yang bisa digunakan untuk mewadahi aktivitas yang berlainan bahkan untuk aktivitas yang sangat bertentangan (seperti aktivitas sakral dan profan) disebut ruang yang relatif.

Ruangan merupakan sebuah objek yang dapat dijadikan sebagai suatu wadah untuk berkespresi bagi sang pengguna karena ruang berkaitan dengan visualisasi dan emosional sang pemilik. Ruangan juga dapat difungsikan sebagai suatu wadah untuk berinteraksi bagi suatu kelompok, misalnya ruang keluarga. Ruang keluarga adalah tempat yang cocok bagi para anggota keluarga untuk berkumpul dan saling berinteraksi pada waktu senggang.

Pada suatu ruangan memiliki ukuran panjang, lebar dan tinggi yang akan menentukan nilai volume dari ruangan tersebut. Besar ukuran setiap ruangan akan berbeda dan menyesuaikan dengan kebutuhan sang pengguna ruangan tersebut.

2.4 Sistem

Sebuah sistem terdiri dari berbagai unsur yang saling melengkapi dalam mencapai tujuan dan sasaran. Unsur-unsur yang terdapat dalam sistem itulah yang disebut dengan subsistem. Subsistem - subsistem tersebut harus saling berhubungan dan berinteraksi melalui komunikasi yang relevan sehingga sistem dapat bekerja secara efektif dan efisien (Eka Iswandy, 2015).

Menurut Tata Sutarbi (2012) yang diacu dalam Rusli (2015) sistem adalah kumpulan elemen - elemen yang saling terkait antara satu dengan yang lain yang tak dapat dipisahkan untuk mencapai satu tujuan tertentu. Secara sederhana, suatu sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau

variabel yang terorganisir, saling berintegrasi, saling tergantung satu sama lain, dan terpadu. Sebuah sistem terdiri atas bagian bagian atau komponen yang terpadu untuk satu tujuan.

Jadi, pada dasarnya sistem merupakan suatu kesatuan dari subsistem atau unsur yang saling berintegrasi dan saling bergantung antara satu sama lain untuk mencapai suatu tujuan tertentu.

2.5 Sistem Kontrol

Sistem kontrol didefinisikan sebagai sekumpulan perangkat yang dirakit untuk membentuk sebuah perangkat gabungan yang dapat menghasilkan sebuah fungsi keluaran spesifik yang diinginkan untuk mengatur sebuah besaran tertentu (Anonim, 2013).

Sistem kontrol adalah sistem yang biasa digunakan sebagai dasar dari pengembangan teknologi modern. Sistem kontrol merupakan sebuah sistem yang dapat mengatur atau mengontrol sesuatu. Sistem kontrol memiliki berapa bagian yang penting, yaitu : masukan (*input*), proses, dan keluaran (*output*). Besaran nilai keluaran (*output*) dalam suatu sistem kontrol adalah hasil manipulasi berdasarkan pengaturan besaran suatu masukan (*input*).

Berdasarkan umpan balik, (*feedback*) sistem kontrol dibedakan menjadi 2 jenis, yaitu :

- a. Sistem kontrol *loop* terbuka (*open loop*)
- b. Sistem kontrol *loop* tertutup (*closed loop*)

a) Sistem Kontrol *Loop* Terbuka (*Open Loop*)

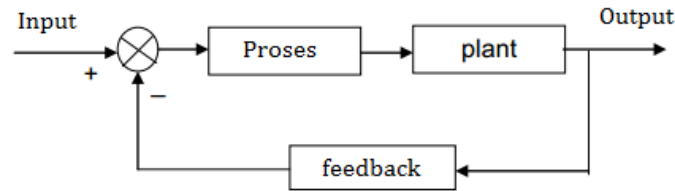
Sistem kontrol *loop* terbuka adalah merupakan suatu proses dalam suatu sistem yang mana variabel input akan berpengaruh pada output yang dihasilkan (Anonim, 2013). Pada sistem kontrol *loop* terbuka, tidak ada umpan balik (*feedback*) informasi yang diberikan dari keluaran (*output*) sehingga kita tidak dapat mengetahui apakah hasil nilai keluaran sudah sesuai dengan yang diinginkan atau belum seperti yang ditunjukkan pada gambar 2.4 berikut.



Gambar 2.4 Sistem Kontrol Terbuka (*Open Loop*)
(Sumber : Dokumentasi Pribadi)

b) Sistem Kontrol *Loop* Tertutup (*Closed Loop*)

Kontrol *loop* tertutup adalah sebuah proses yang mana variabel yang dikontrol secara terus menerus disensor kemudian dibandingkan dengan kuantitas referensi (Anonim, 2013). Sistem kontrol *loop* tertutup (*closed loop*) memiliki nilai umpan balik (*feedback*) dari suatu keluaran sehingga dapat diketahui hasil keluaran tersebut apakah telah sesuai dengan nilai yang diinginkan atau belum seperti yang ditunjukkan pada gambar 2.5 berikut.



Gambar 2.5 Sistem Kontrol Tertutup (*Closed Loop*)
(Sumber : dokumentasi pribadi)

2.6 Logika Fuzzy

Logika *Fuzzy* merupakan suatu logika yang memiliki nilai kekaburan atau kesamaran (*fuzzyness*) antara benar atau salah. Dalam teori logika *fuzzy* suatu nilai bisa bernilai benar atau salah secara bersama. Namun berapa besar keberadaan dan kesalahan suatu tergantung pada bobot keanggotaan yang dimilikinya. Logika *fuzzy* memiliki derajat keanggotaan dalam rentang 0 hingga 1. Berbeda dengan logika digital yang hanya memiliki dua nilai 1 atau 0. Logika *fuzzy* digunakan untuk menerjemahkan suatu besaran yang diekspresikan menggunakan bahasa (*linguistic*), misalkan besaran kecepatan laju kendaraan yang diekspresikan dengan pelan, agak cepat, cepat, dan sangat cepat. Dan logika *fuzzy* menunjukkan sejauh mana suatu nilai itu benar dan sejauh mana suatu nilai itu salah (Helfi Nasution, 2012).

Logika *fuzzy* merupakan suatu metode yang banyak diterapkan dalam pengembangan teknologi terbaru karena konsep logika *fuzzy* lebih mudah untuk dimengerti dan memiliki pola yang meniru pola pikir manusia. Metode logika *fuzzy* pertama kali diperkenalkan oleh Dr. Lotfi Zadeh dari Universitas California, Berkeley pada 1965. Logika *fuzzy* dan logika probabilitas secara matematis sama keduanya mempunyai nilai kebenaran yang berkisar antara 0 dan 1 namun secara konsep berbeda.

Logika *fuzzy* berbicara mengenai "derajat kebenaran", sedangkan logika probabilitas mengenai "probabilitas, kecenderungan". Karena kedua hal itu berbeda, Logika *fuzzy* dan logika probabilitas mempunyai contoh penerapan dalam dunia nyata yang berbeda. Proses pengolahan data dalam metode logika *fuzzy* meliputi beberapa tahap, yaitu :

1. *Fuzzification*, merupakan proses awal dari suatu logika *fuzzy*. Pada tahap *fuzzification* suatu nilai *crisp* (0 atau 1) dirubah menjadi suatu *input fuzzy* (antara 0 dan 1).
2. *Inference Rule Based*, merupakan proses kedua dari suatu logika *fuzzy*. Tahap ini adalah tahap pengambilan keputusan serta pemberian aturan (*rule*) terhadap suatu *input fuzzy* yang akan menghasilkan suatu *output fuzzy*.
3. *Defuzzification*, merupakan proses terakhir dari suatu logika *fuzzy*. Tahap ini adalah tahap dimana suatu nilai *output fuzzy* yang telah didapatkan berdasarkan perhitungan dengan suatu metode tertentu pada proses *Inference Rule Based* diubah kembali menjadi suatu nilai *crisp*.

Bentuk diagram blok dari proses pengolahan data dengan logika *fuzzy* dapat dilihat pada gambar 2.6 berikut.



Gambar 2.6 Diagram Blok Logika *Fuzzy*

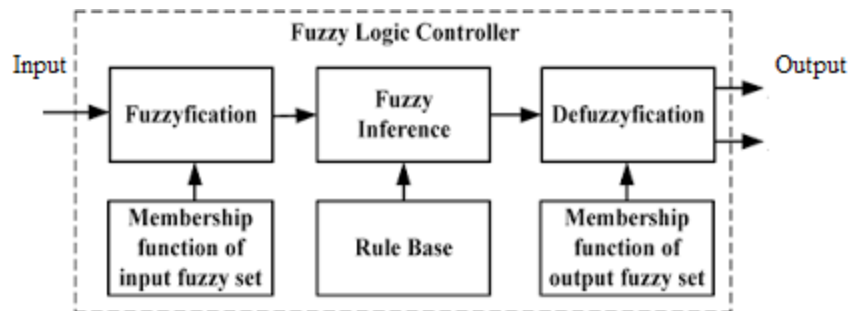
(Sumber : <http://saintisislami.blogspot.co.id/2011/06/blok-diagram-fuzzy-logic-control.html>)

Didalam logika *fuzzy* terdapat beberapa metode penalaran yang digunakan pada proses inferensi ataupun pemberian aturan pada suatu *input fuzzy*, antara lain : metode mamdani, metode sugeno, metode tsukamoto, dan sebagainya.

Penalaran dengan metode mamdani mirip dengan metode sugeno, namun terjadi perbedaan pada proses defuzzyfikasi dimana pada metode mamdani menggunakan beberapa himpunan *fuzzy* yang diambil secara acak sedangkan pada metode sugeno berupa konstanta atau persamaan linier. Metode sugeno diperkenalkan oleh Takagi Sugeno-Kang pada tahun 1985 yang merupakan suatu metode perbaikan dari sistem *fuzzy* murni dengan menambahkan suatu model matematika sederhana sebagai bagian *THEN*. Pada sistem *fuzzy* sugeno, sistem *fuzzy* memiliki suatu nilai rata-rata tertimbang (*Weighted Average Values*) di dalam bagian aturan *fuzzy IF-THEN*.

2.7 Fuzzy Logic Controller

Fuzzy Logic Controller sudah banyak digunakan dalam kehidupan sehari-hari. Banyak barang yang telah diciptakan dengan menggunakan *Fuzzy Logic Controller* sebagai pengendalinya, seperti : *rice cooker*, mesin cuci, *vacuum cleaner*, dan termos pemanas air. Selain itu, *Fuzzy Logic Controller* juga diaplikasikan kedalam sistem transportasi seperti *elevator*, sistem kendali lalu lintas ataupun dalam bentuk perangkat lunak (*software*) yang dimanfaatkan untuk pengolahan data. Bentuk diagram dari *Fuzzy Logic Controller* dapat dilihat pada gambar 2.7 berikut.



Gambar 2.7 Diagram *Fuzzy Logic Controller*

(Sumber : <http://kursuselektronikaku.blogspot.co.id/2015/09/membuat-alat-kendali-motor-dc-dengan.html>)

Fuzzy Logic Controller banyak digunakan dalam perkembangan teknologi pada masa kini karena memiliki beberapa kelebihan, diantaranya :

- a) *Fuzzy Logic Controller* memiliki daerah kerja yang lebih luas sehingga meminimalisir nilai kesalahan (*error*)
- b) *Fuzzy Logic Controller* merupakan sistem kontrol yang fleksibel dan dapat memberikan toleransi terhadap suatu data yang tidak tepat.

2.8 Pemrograman MATLAB

Menurut Naba (2009:26) MATLAB adalah bahasa pemrograman tingkat tinggi dimana arti perintah dan fungsi-fungsinya bisa dimengerti dengan mudah, meskipun bagi seorang pemula. Hal itu karena di dalam MATLAB, masalah dan solusi bisa diekspresikan dalam notasi-notasi matematis yang bisa dipakai. MATLAB singkatan dari *Matrix Laboratory*. Pada awalnya MATLAB dimaksudkan sesuai dengan namanya, yaitu untuk menangani berbagai operasi matriks dan vektor menggunakan rutin-rutin dan *library* LINPACK dan EISPACK. Saat ini MATLAB telah menggabungkan rutin-rutin dan *library* dari LAPACK dan BLAS, yang lebih efisien dalam menangani operasi matriks dan vektor.

Sebuah *toolbox* dalam MATLAB adalah koleksi berbagai fungsi MATLAB (M-Files, yaitu file berekstensi), yang merupakan perluasan MATLAB untuk memecahkan masalah-masalah khusus bidang tertentu. Beberapa bidang sudah tersedia *toolbox*-nya dalam MATLAB, meliputi *Fuzzy Logic*, *neural network* (jaringan syaraf tiruan), *control system* (sistem kontrol), *signal processing* (pengolahan sinyal), dan *wavelet*. MATLAB adalah singkatan dari *MATRIX LABORATORY*, yang biasanya di gunakan dalam :

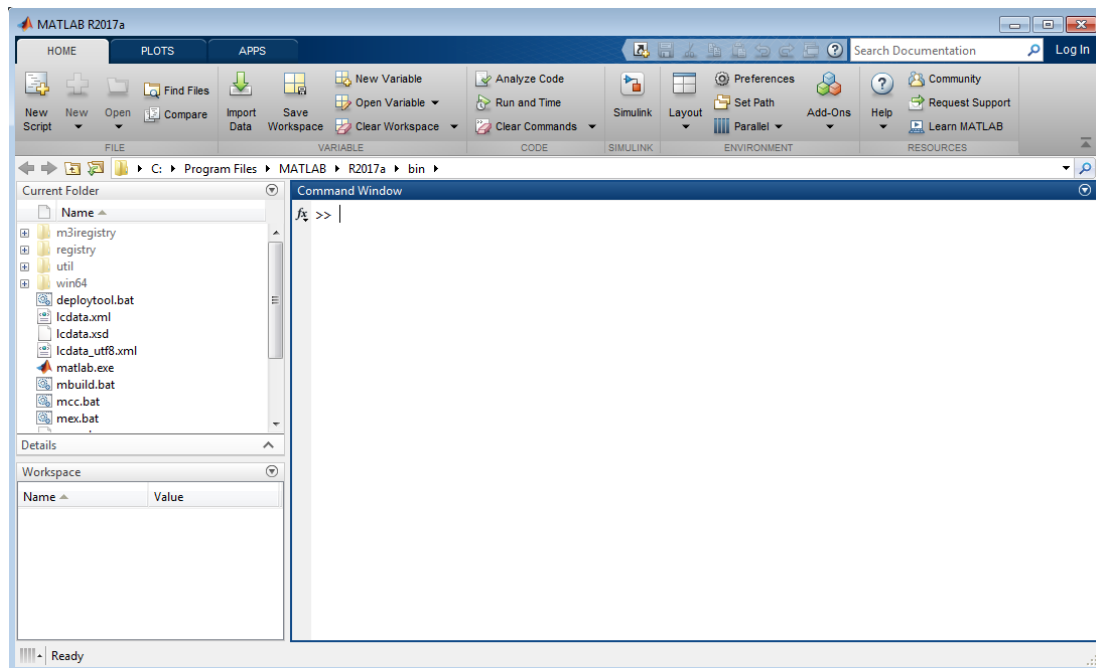
- a) Pengembangan Algoritma matematika dan komputasi
- b) Pemodelan, simulasi, dan pembuatan *prototype* dari penerimaan data
- c) Analisa, eksplorasi, dan visualisasi data
- d) *Scientific* dan *engineering*

e) Pengembangan aplikasi berbasis grafik dan pembuatan *Graphical User Interface* (GUI).

Secara *default*, MATLAB terdiri dari :

1. *Command window* yang merupakan tempat dimana user menuliskan fungsi yang diinginkan.
2. *Command history* untuk melihat dan menggunakan kembali fungsi-fungsi sebelumnya.
3. *Workspace* yang berisi variabel yang digunakan dan untuk membuat variabel baru dalam MATLAB.
4. *Current directory* menunjukkan folder-folder yang berisi file MATLAB yang sedang berjalan.

Tampilan awal *software* MATLAB dapat dilihat pada gambar 2.8 berikut.



Gambar 2.8 Tampilan Awal MATLAB

(Sumber : dokumentasi pribadi)

2.9 Fuzzy Logic Menggunakan MATLAB

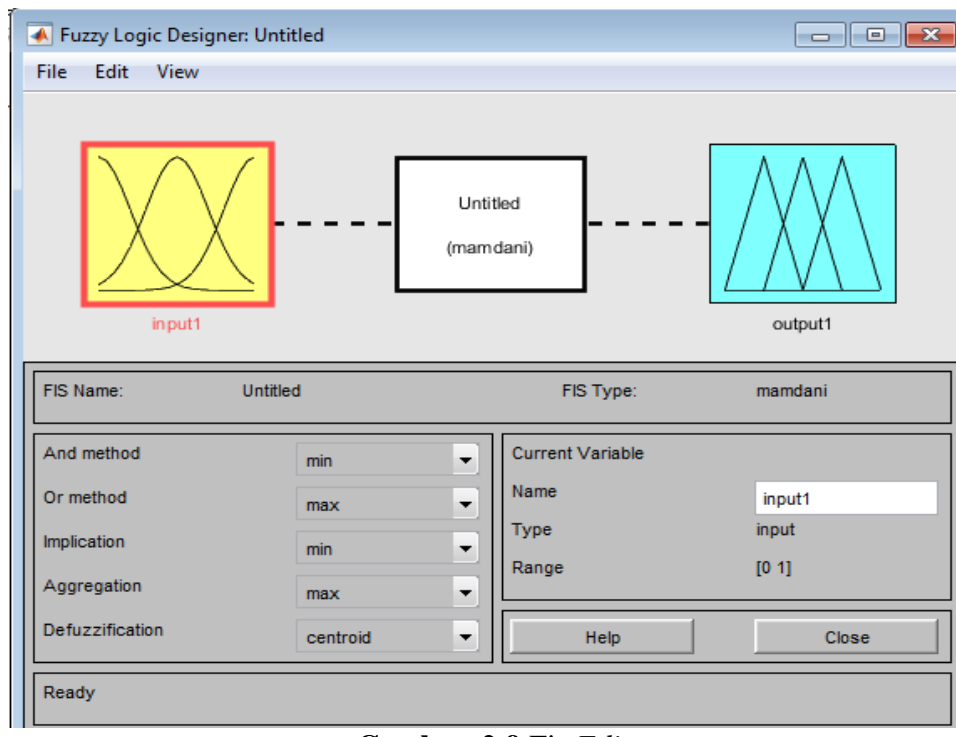
Menurut Naba (2009:33) yang diacu dalam Maudi (2016) MATLAB adalah bahasa pemrograman teknis yang sangat baik untuk proses simulasi dan visualisasi data. MATLAB menjadi *tools* bagi para ilmuwan dan pelajar untuk memodelkan sistem, menganalisis serta menampilkan data. Saat ini, MATLAB terus berkembang untuk menyokong berbagai disiplin keilmuan, termasuk Logika *Fuzzy*. MATLAB telah menyediakan sebuah *tools* untuk merancang logika *fuzzy*, yang dikenal sebagai *Fuzzy Logic Toolbox*. *Fuzzy Logic toolbox* adalah sekumpulan *tool* yang akan membantu *user* merancang sistem *fuzzy* untuk diaplikasikan dalam berbagai bidang, seperti *automatic control*, *signal processing*, *identification system*, *pattern recognitions*, *time series prediction*, *data mining*, dan bahkan *financial applications*. Dengan *Fuzzy Logic toolbox*, *user* bisa membuat atau meng-*edit* FIS dalam lingkungan kerja MATLAB. *User* bahkan dapat membangun FIS *stand-alone* dalam bahasa C yang akan memanggil FIS yang *user* bangun dalam lingkungan MATLAB.

Fuzzy Logic Toolbox sangat bersahabat bagi pengguna, memungkinkan *user* untuk berkreasi dengan bebas dalam rancang bangun FIS. Contohnya, *user* bisa mengganti fungsi-fungsi bawaan (*default*) MATLAB yang dipakai dalam lima tahap pembangunan FIS dengan fungsi buatan *user* sendiri : fungsi keanggotaan, *operator AND*, *operator OR*, metode implikasi, metode agregasi, dan metode defuzzyfikasi.

Semua *tool* dalam *Fuzzy Logic toolbox* dikelompokkan menjadi tiga kategori yaitu *command line*, *graphical user interface (GUI)*, dan *simulink block*. Fungsi-fungsi *command lines Fuzzy Logic toolbox* adalah fungsi-fungsi yang dapat dieksekusi dari MATLAB *Prompt*. Sebagian besar fungsi ini ditulis dalam bentuk *M-Files*.

2.9.1 FIS Editor

Naba (2009:37) yang diacu dalam Maudi (2016) Pada *MATLAB Promt*, ketikkan : **Fuzzy** , Maka muncul *FIS Editor* seperti tampak pada gambar 2.9 dengan variabel masukan berlabel **input1** dan variabel keluaran berlabel **output1**.



Gambar 2.9 Fis Editor

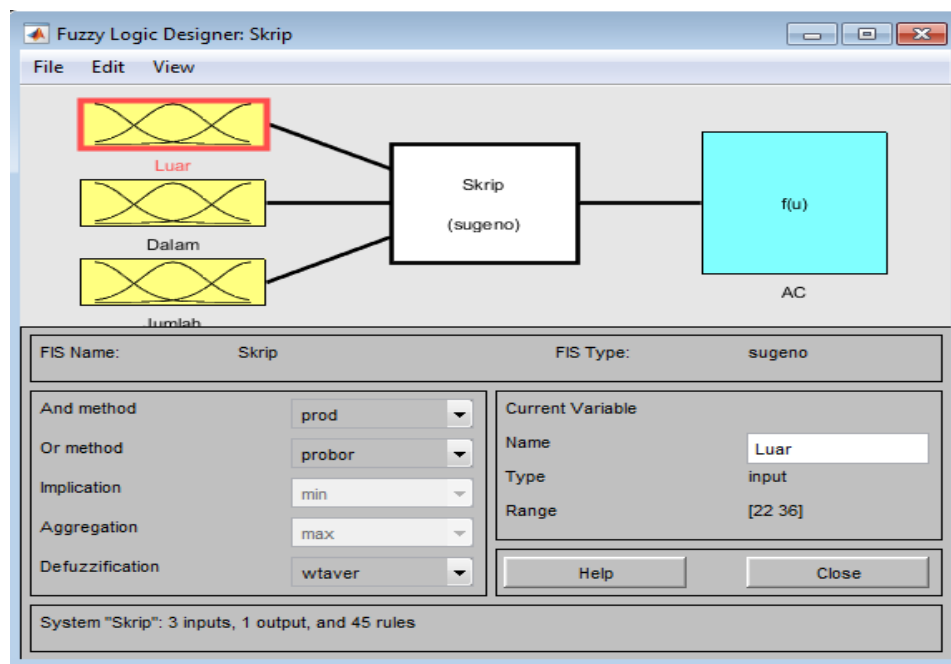
(Sumber : dokumentasi pribadi)

Untuk contoh kasus yang akan dibahas membuat FIS dengan tiga masukan : **suhu udara luar, suhu udara didalam ruangan** serta **jumlah manusia** dan keluaran : **Setting AC**. Langkah-langkahnya adalah sebagai berikut, pertama pilih menu **Edit**, kemudian pilih **Add Variable**, lalu pilih **Add Input** sehingga muncul kotak kuning batu dengan label **input2** yang mana akan menjadi masukan kedua dari FIS, lalu ubah label **input1** menjadi **Suhu Udara Luar** dengan mengklik kotak kuning berlabel **input1** (saat diklik, batas kotak akan berubah warna menjadi merah), lalu pada bagian **Current**

Variable di sebelah kanan bawah, pada kolom nama, ganti label variabel *input1* menjadi **Suhu Udara Luar** dan tekan *Enter*.

Lakukan hal serupa dengan langkah sebelumnya untuk mengubah label *input2* menjadi **Suhu Dalam** dan label *input3* menjadi **Jumlah Manusia**. Ubah label *output1* menjadi **Setting AC** dengan mengklik kotak biru di kanan atas (saat diklik, batas kotak akan berubah warna menjadi merah), lalu pada bagian *Current Variable* di sebelah kanan bawah, pada kolom nama, ganti label variabel *output1* menjadi **Setting AC** dan tekan *Enter*.

Kemudian pilih menu File, pilih *Export*, dan pilih To *Workspace*, pada *edit Field Workspace variable*, ketik **Skrip** lalu klik **OK**. Langkah ini akan menyimpan FIS ke dalam variabel *Workspace* bernama Skrip. Gambar 2.10 merupakan tampilan dari diagram FIS yang telah dibuat.



Gambar 2.10 Diagram Fis AC

(Sumber : dokumentasi pribadi)

2.9.2 *Membership Function Editor*

Fungsi-fungsi keanggotaan variabel masukan dan keluaran didefinisikan melalui *Membership Function Editor*. *Membership Function Editor* merupakan sebuah fitur yang digunakan untuk mengatur nilai dari fungsi-fungsi keanggotaan *fuzzy* agar dapat disesuaikan dengan parameter yang diinginkan oleh pembuatnya.

Ada tiga cara untuk membuka *Membership Function Editor* :

1. Dari **FIS Editor**, pilih *View*, lalu pilih *Edit Membership Functions*.
2. Dari **FIS Editor**, klik ikon variabel masukan/keluaran.
3. Dari **MATLAB Command Window**, ketikkan **mfedit Skrip**, mengasumsikan bahwa **user** sudah terlanjur menutup **FIS Editor**.

Fitur-fitur dalam *Membership Function Editor* serupa dengan fitur-fitur dalam *FIS Editor*, dan juga semua GUI FIS yang belum disebut sejauh ini. Dengan *Membership Function Editor*, **user** bisa menampilkan dan meng-*edit* semua fungsi keanggotaan dari variabel FIS baik variabel masukan (*input*) ataupun variabel keluaran (*output*).

2.10 *Arduino Uno R3*

Arduino Uno R3 merupakan sebuah *minimum system* yang digunakan sebagai pusat pengolahan data serta kendali untuk suatu sistem kontrol. *Arduino Uno R3* merupakan jenis sistem mikrokontroler yang relatif mudah dan cepat dalam membuat aplikasi elektronika maupun robotika.

Arduino Uno R3 terdiri dari perangkat elektronik atau papan rangkaian elektronik *open source* yang di dalamnya terdapat komponen utama yaitu sebuah *chip* mikrokontroler dengan jenis AVR dari perusahaan Atmel berbasis *ATmega328*. *Arduino Uno R3* memiliki 14 *digital pin input/output*, seperti pada gambar 2.11, dimana 6 *pin* digunakan sebagai *output* PWM, 6 *pin input analog*, 16 MHz resonator keramik, koneksi USB, jack catu daya eksternal, *header ICSP*, dan tombol *reset*.. *Datasheet* dari *Arduino Uno R3* dapat dilihat pada Lampiran 5.1.



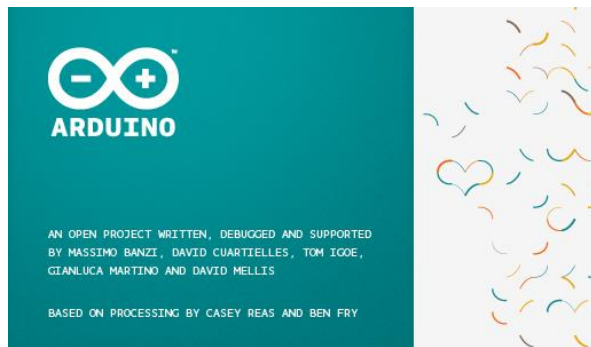
Gambar 2.11 Arduino Uno R3

(Sumber : <https://store.arduino.cc/usa/arduino-uno-rev3>)

2.11 Arduino IDE

IDE (*Integrated Development Environment*) *Arduino* merupakan aplikasi yang mencakup *editor*, *compiler*, dan *uploader* dapat menggunakan semua seri modul keluarga *arduino* seperti *Arduino Duemilanove*, *Uno*, *Bluetooth*, *Mega* .

Aplikasi *Arduino IDE* merupakan aplikasi yang bersifat *open source* sehingga memungkinkan penggunaanya untuk melakukan modifikasi dan pengembangan. Aplikasi *Arduino IDE* memiliki banyak versi yang dapat diunduh pada website resminya. Aplikasi yang digunakan dalam penelitian ini adalah aplikasi *Arduino IDE* versi 1.8.0. *Form splash* dari *arduino 1.8.0* diperlihatkan pada gambar 2.12 berikut ini.



Gambar 2.12 *Form Splash Arduino IDE 1.8.0*
(Sumber : dokumentasi pribadi)

2.12 *Thermocouple*

Thermocouple merupakan suatu sensor yang digunakan untuk mendeteksi suatu suhu melalui dua jenis konduktor berbeda yang berada pada ujung sensor sehingga terjadi efek *thermo-electric*. Sensor ini menjadi sensor yang populer dan sering digunakan untuk pendeteksi suhu karena memiliki nilai rentang suhu yang luas dan memiliki respon yang sangat cepat terhadap setiap perubahan suhu yang terjadi. Umumnya, sensor *thermocouple* dapat mendeteksi nilai temperatur mulai dari $-200\text{ }^{\circ}\text{C}$ sampai $1350\text{ }^{\circ}\text{C}$ (untuk tipe K) dan dapat mencapai di atas $2300\text{ }^{\circ}\text{C}$ untuk beberapa tipe yang lain. Bentuk dari sensor *thermocouple* dapat dilihat pada gambar 2.13 berikut ini.



Gambar 2.13 *Thermocouple*
(Sumber : <https://www.tokopedia.com/diggysshop/thermocouple-k-type-sensor-module-temperature-01024c>)

2.13 *Push Button*

Push button atau saklar tekan berfungsi sebagai terminal *wiring* yang dihubungkan dengan alat listrik lainnya, mempunyai kapasitas beban sekitar 5A. Prinsip kerja *Push Button* adalah apabila dalam keadaan normal tidak ditekan maka kontak tidak berubah, apabila ditekan maka kontak NC akan berfungsi sebagai *stop* (memberhentikan) dan kontak NO akan berfungsi sebagai *start* (menjalankan) biasanya digunakan pada sistem pengontrolan motor – motor induksi untuk menjalankan atau mematikan motor pada industri – industri. Bentuk dan simbol dari *push button* dapat dilihat pada gambar 2.14 berikut.

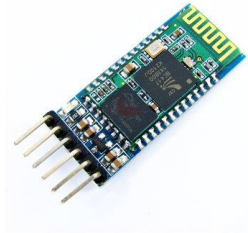


Gambar 2.14 *Push Button*

(Sumber : <https://www.sparkfun.com/products/97>)

2.14 *HC-05 Bluetooth Module*

HC-05 bluetooth module merupakan sebuah modul komunikasi nirkabel pada frekuensi 2,4 GHz yang memiliki komunikasi SPP (*Serial Port Protocol*) dan cukup mudah untuk diintegrasikan dengan berbagai macam mikrokontroler. *HC-05* dapat bekerja pada tegangan 3,6 - 6 volt DC. Jarak untuk komunikasinya berkisar antara 5 – 10 meter. Lebih dari 10 meter akan menyebabkan kualitas koneksi dari modul *HC-05* akan berkurang. Bentuk dari *HC-05 bluetooth module* dapat dilihat pada gambar 2.15 berikut.



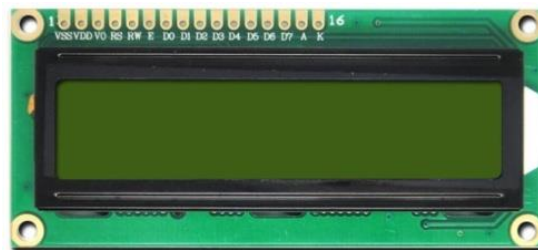
Gambar 2.15 *HC-05 Bluetooth Module*

(Sumber : <https://forum.arduino.cc/index.php?topic=334629.0>)

2.15 LCD (Liquid Crystal Display) 16x2

LCD (Liquid Crystal Display) adalah suatu *display* dari bahan cairan kristal yang pengoperasiannya menggunakan sistem *dot* matriks. *LCD* banyak digunakan sebagai *display* dari alat-alat elektronik, seperti pada kalkulator, jam digital, multimeter digital, dan lain sebagainya. *LCD 16x2* berarti *LCD* tersebut memiliki tampilan dengan lebar 16 kolom dan 2 baris atau biasa disebut sebagai *LCD Karakter 16x2* (Anggi R., 2016).

LCD (Liquid Crystal Display) dapat menampilkan karakter berupa huruf, angka ataupun simbol. Modul *LCD* telah dilengkapi dengan memori dan kontroler yang memiliki dua *register* instruksi (IR) dan *register* data (DR). Gambar dari *LCD 16x2* dapat dilihat pada gambar 2.16.



Gambar 2.16 *LCD 16x2*

(Sumber : https://www.makerfabs.com/index.php?route=product/product&product_id=140)

LCD ini memiliki 16 *pin* konektor (didefinisikan pada tabel 2.1)

Tabel 2.1 Fungsi dan Konfigurasi Pin-Pin LCD

Pin	Nama	Fungsi
1	VSS	<i>Ground</i>
2	VDD	+5V
3	VO	Pengaturan Kontras
4	RS	<i>Register Select Signal</i> (0 = mode tulis, 1 = mode baca)
5	R/W	<i>Read/Write Signal</i> (memilih mode tulis atau baca)
6	E	<i>Enable</i>
7	DB0	<i>H/L Data Bus Line</i>
8	DB1	<i>H/L Data Bus Line</i>
9	DB2	<i>H/L Data Bus Line</i>
10	DB3	<i>H/L Data Bus Line</i>
11	DB4	<i>H/L Data Bus Line</i>
12	DB5	<i>H/L Data Bus Line</i>
13	DB6	<i>H/L Data Bus Line</i>
14	DB7	<i>H/L Data Bus Line</i>
15	A	<i>Negative Voltage Output</i>
16	K	<i>Power Supply for B/L (OV)</i>

2.16 Penelitian Yang Relevan

Pada penelitian sebelumnya yang ditulis oleh Maudi Mayangsari (2016) dengan judul penelitian *Analisis Fuzzy Logic* Pada Pengaruh Suhu Udara Luar Dan Jumlah Manusia Terhadap Suhu *Setting Air Conditioner* (AC) Di Gedung L1 Jurusan Teknik Elektro Fakultas Teknik Universitas Negeri Jakarta, Dengan membuat model *Fuzzy Logic* dan suhu *setting Air Conditioner* (AC) sebagai keluarannya dan kemudian menjadikan keluaran tersebut sebagai referensi dalam simulasi yang membandingkan

besar suhu *setting Air Conditioner* (AC) dengan suhu aktual ruangan yang diambil melalui pengukuran.

Hasil yang didapat adalah besaran suhu *setting Air Conditioner* (AC) yang sesuai dengan suhu udara luar dan jumlah manusia pada saat pemakaian yang diuji dengan simulasi dari suhu *setting Air Conditioner* (AC). Dapat terlihat pengaruh suhu udara luar terhadap besar suhu *setting Air Conditioner* (AC) pada semakin meningkatnya suhu udara luar maka semakin rendah besar suhu *setting Air Conditioner* (AC) dan semakin besar juga *range* suhu *setting Air Conditioner* (AC) dengan suhu aktual ruangan. Pengaruh jumlah manusia terhadap suhu *setting Air Conditioner* (AC) dapat terlihat pada semakin padat jumlah manusia dalam suatu ruang maka semakin rendah besar suhu *setting Air Conditioner* (AC) dan semakin besar juga *range* suhu *setting Air Conditioner* (AC) dengan suhu aktual ruangan.

Berdasarkan, hasil penelitian didapatkan model *Fuzzy Logic* yang dibuat peneliti efektif untuk digunakan, hal ini terbukti melalui simulasi yang dilakukan peneliti. Dan pada penelitian selanjutnya adalah pengembangan dari analisis yang dilakukan oleh Maudi Mayangsari (2016) yaitu membuat suatu penerapan metode kontrol logika *fuzzy* melalui suatu *prototype* alat yang menggunakan beberapa komponen dasar yaitu : sensor suhu *thermocouple* sebagai sensor yang berfungsi untuk membaca nilai suhu baik didalam maupun diluar ruangan, *push button* yang berfungsi untuk menghitung jumlah penghuni yang masuk ataupun keluar ruangan dan output yang berupa *Automatic Remote Control* yang akan mengatur nilai suhu pada *Air Conditioner* (AC) sebagai sebuah sistem yang menjadi realisasi dari analisis yang dilakukan pada penelitian sebelumnya.

2.17 Kerangka Berpikir

AC (*Air Conditioner*) memiliki peran yang cukup besar dalam menunjang kegiatan manusia di dalam suatu ruangan. AC (*Air Conditioner*) berfungsi untuk mengkondisikan suhu udara di dalam suatu ruangan agar kondisi menjadi lebih nyaman. Namun, seringkali terjadi permasalahan pada pengoperasian dari suatu AC (*Air Conditioner*) dimana pengaturan suhu pada AC (*Air Conditioner*) tersebut tidak diatur sesuai dengan kondisi ruangan yang menyebabkan kinerja AC (*Air Conditioner*) menjadi tidak maksimal.

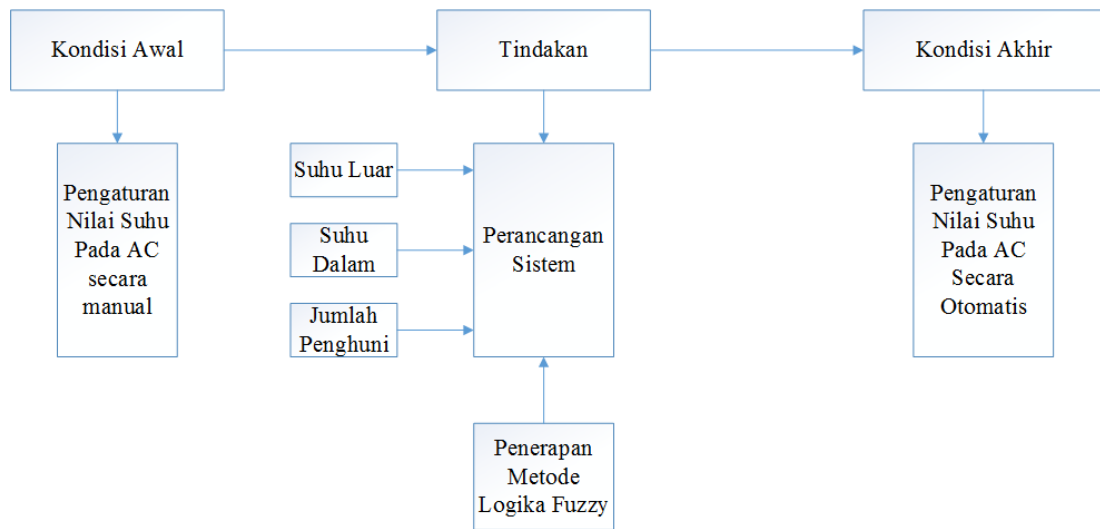
Terdapat beberapa faktor yang dapat mempengaruhi kondisi suhu pada suatu ruangan, diantaranya adalah jumlah manusia yang ada didalam ruangan tersebut dan pengaruh suhu udara luar yang menyebabkan terjadinya peningkatan atau penurunan suhu di dalam ruangan tersebut. Kondisi suhu ruangan terendah biasanya akan terjadi ketika suhu udara luar berada pada tingkat dingin ($< 20^{\circ}\text{C}$) dan jumlah penghuni ruangan sedikit (< 5 orang) sehingga AC (*Air Conditioner*) harus disesuaikan dengan *setting* suhu yang cukup tinggi. Kondisi suhu ruangan tertinggi biasanya terjadi ketika suhu udara luar berada pada tingkat panas ($> 34^{\circ}\text{C}$) dan jumlah penghuni ruangan sangat padat (mencapai 40 orang) sehingga AC (*Air Conditioner*) harus disesuaikan dengan *setting* suhu yang rendah agar kondisi didalam ruangan menjadi lebih nyaman.

Untuk mendapatkan suatu perhitungan yang tepat untuk menentukan *setting* suhu yang maksimal dari AC (*Air Conditioner*) berdasarkan kondisi-kondisi yang terjadi maka dibutuhkan suatu sistem pengaturan yang mampu memetakan berbagai kondisi yang berbeda berdasarkan pengaruh suhu udara luar dan jumlah manusia yang ada di dalam ruangan tersebut. Logika *Fuzzy* merupakan metode sistem pengaturan yang

cukup baik dan dianggap memiliki keakuratan dalam mengolah data-data yang samar menjadi suatu nilai yang pasti karena logika *fuzzy* mampu memetakan berbagai situasi dan kondisi hingga mendapatkan hasil yang sesuai dengan yang diinginkan.

Menggunakan suhu udara luar, suhu udara dalam ruangan dan jumlah penghuni ruangan sebagai masukan dan mengelompokan serta mengolah data masukan tersebut menggunakan logika *fuzzy* akan menghasilkan suatu *output* berupa pengelompokan nilai *setting* suhu yang membuat tingkat pendinginan AC (*Air Conditioner*) dapat diatur sesuai dengan hasil yang diperoleh dari perhitungan logika *fuzzy*. Metode logika *fuzzy* diterapkan secara langsung kedalam sistem kendali *Arduino Uno R3* melalui kode program yang dibuat oleh peneliti. Penerapan logika *fuzzy* kedalam sistem kendali *Arduino Uno R3* dilakukan agar pengaturan tingkat pendinginan AC (*Air Conditioner*) dapat terjadi secara otomatis.

Kerangka berpikir ini dimaksudkan untuk memberi pengetahuan mengenai alat yang dibuat peneliti beserta metode logika *fuzzy* yang diterapkan kedalam sistem kendali dengan variabel masukan kondisi suhu udara luar, kondisi suhu udara dalam ruangan dan jumlah penghuni ruangan dengan variabel keluaran berupa nilai *setting* suhu pada AC (*Air Conditioner*). Diagram blok *kerangka* berpikir dapat dilihat pada gambar 2.17 berikut.



Gambar 2.17 Diagram Blok Kerangka Berpikir
(Sumber : dokumentasi pribadi)

BAB III

METODE PENELITIAN

3.1 Waktu dan Tempat Penelitian

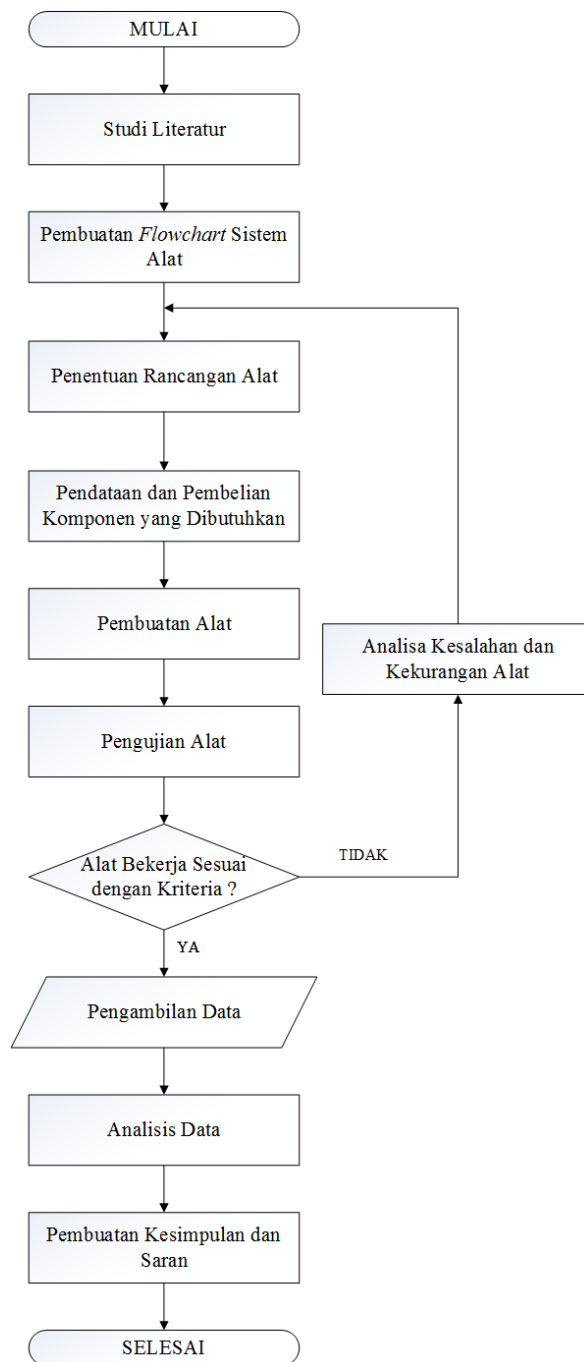
Penelitian dilakukan di Lantai 1 Ruang Laboratorium Pengukuran Listrik, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Negeri Jakarta dan di rumah yang beralamat di Jalan Gongseng Raya RT.06/09, Kelurahan Baru, Kecamatan Pasar Rebo, Jakarta Timur. Dalam rentang waktu pada bulan Maret – Juni 2017. Tujuan dari penelitian adalah untuk membuat sebuah *prototype* alat yang dapat mengatur suhu pada *Air Conditioner* (AC) secara otomatis berdasarkan kondisi suhu udara luar, suhu udara didalam ruangan dan jumlah orang yang ada didalam ruangan tersebut.

3.2 Alat dan Bahan Penelitian

1. *Notebook* yang digunakan dalam penelitian untuk pembuatan laporan dan program dengan spesifikasi sebagai berikut:
 - a. AMD A4-5000 APU with Radeon™ HD *Graphics* 1.50 GHz
 - b. *Memory* 4.00 GB (3.74 GB *usable*)
 - c. *Solid State Drive* (SSD) 500 GB
 - d. Sistem Operasi *Windows 7 Ultimate* 64 bit
2. *Software* pendukung terdiri dari:
 - a. *Arduino IDE* 1.8.0
 - b. *PCB Wizard* 3.5.0
 - c. *Microsoft Word* 2016
 - d. *Google Sketchup* 2017

- e. MATLAB R2017b
 - f. *Snipping Tool*
 - g. *Microsoft Visio 2016*
 - h. *Proteus 8 Professional*
3. *Hardware* pendukung yang digunakan terdiri dari:
- a. Solder listrik
 - b. *Mini electric drill* (Bor tangan kecil)
 - c. Gunting
 - d. Tang potong
 - e. *Double Tape*
 - f. *Multimeter Digital*
 - g. Gerinda
 - h. Tang Jepit
 - i. Obeng set
 - j. Spidol Permanen
 - k. Bor Duduk
 - l. Kikir
 - m. *Glue Gun*

3.3 Diagram Alir Penelitian



Gambar 3.1 *Flowchart* Penelitian
(Sumber : dokumentasi pribadi)

3.3.1 Studi Literatur

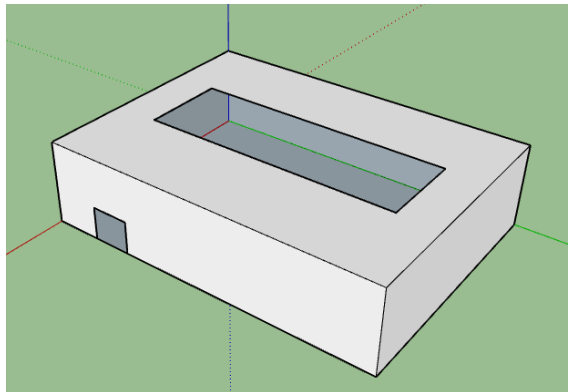
Tahap pertama pada penelitian adalah studi literatur. Studi literatur dilakukan untuk mencari informasi terkait penelitian yang akan dilakukan, mulai dari teori tentang logika *fuzzy* dan komponen-komponen yang dibutuhkan untuk membuat sebuah *prototype* sistem pengaturan suhu *Air Conditioner* (AC) secara otomatis.

3.3.2 Pembuatan *Flowchart* Sistem Kerja Alat

Tahap selanjutnya adalah pembuatan diagram alir (*flowchart*) dari sistem yang akan dibuat. Pembuatan diagram alir (*flowchart*) dari sistem dilakukan untuk mengetahui cara kerja alat tahap demi tahap. Gambar *flowchart* sistem kerja alat dapat dilihat pada lampiran 5.3.

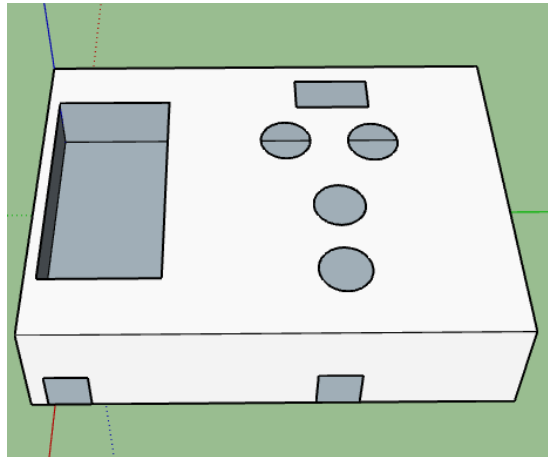
3.3.3 Penentuan Rancangan Alat

Setelah pembuatan diagram alir (*flowchart*), dilanjutkan dengan penentuan desain rancangan alat. Desain yang akan dibuat adalah desain *box remote control receiver* dan *box remote control transmitter*. Pembuatan desain rancangan alat dilakukan dengan menggunakan *software Google Sketchup 2017* dan dibuat dalam bentuk desain 3D. Gambar 3.2 dan gambar 3.3 merupakan gambar desain dari alat yang dibuat.



Gambar 3.2 Desain *Box Remote Control Receiver*

(Sumber : dokumentasi pribadi)



Gambar 3.3 Desain *Box Remote Control Transmitter*

(Sumber : dokumentasi pribadi)

3.3.4 Pendataan dan Pembelian Komponen Dan Bahan Yang Dibutuhkan

Setelah pembuatan rancangan alat, selanjutnya adalah pendataan serta pembelian komponen yang dibutuhkan. Setelah melakukan studi literatur, telah didapatkan informasi ataupun data terkait komponen-komponen yang dibutuhkan dalam pembuatan alat. Setelah seluruh komponen didata, tahap selanjutnya adalah pembelian komponen-komponen yang dibutuhkan. Tabel 3.1 adalah daftar komponen yang dibutuhkan dalam pembuatan alat.

Tabel 3.1 Daftar Komponen Yang Dibutuhkan

No	Nama Komponen	Jenis	Jumlah
1	<i>Arduino</i>	<i>Uno R3</i>	2
2	Sensor Suhu	Thermocouple Type K	2
3	<i>Push Button</i>	-	4
4	<i>LCD 16x2</i>	-	1
5	<i>Bluetooth Module</i>	HC-05	2
6	Relay 5 VDC	-	3
7	<i>Black Box</i>	X5	2

3.3.5 Pembuatan Alat

Setelah seluruh komponen didapatkan, tahap berikutnya adalah pembuatan alat. Pada tahap ini dilakukan pembuatan sistem secara keseluruhan baik pembuatan *hardware* dan pembuatan *software*.

3.3.6 Pengujian alat

Setelah alat selesai dibuat, dilakukan pengujian untuk melihat apakah alat bekerja sesuai dengan kriteria atau tidak. Jika alat bekerja sesuai dengan kriteria, maka dilanjutkan ke tahap pengambilan data. Jika alat belum bekerja sesuai dengan kriteria, maka dilakukan analisa kesalahan dan kekurangan alat untuk mengetahui kesalahan atau kekurangan dari alat yang dibuat, lalu kembali ke tahap perancangan. Beberapa kriteria yang dibuat adalah sebagai berikut:

- a. Sensor suhu dapat mengukur nilai suhu dengan waktu pembacaan setiap satu detik sekali
- b. *Push Button* dapat bekerja dan memberikan nilai logika yang sesuai dengan keadaan *push button* (ditekan/tidak ditekan) dan tidak terjadi *bouncing*.
- c. *Bluetooth* dapat terkoneksi dan dapat mengirim ataupun menerima data dengan baik.
- d. Kontroler mampu mengolah data input dengan perhitungan logika *fuzzy* dan menampilkan hasilnya pada layar *LCD*.

3.3.7 Pengambilan Data

Setelah alat diuji dan sesuai dengan kriteria, maka dilanjutkan ke tahap pengambilan data. Pada tahap ini, dilakukan pengukuran dari setiap komponen yang digunakan dan mencatat hasilnya.

3.3.8 Analisis Data

Setelah data diambil, dilakukan analisis data untuk mendapatkan informasi terkait alat yang telah dibuat berdasarkan data yang telah diambil.

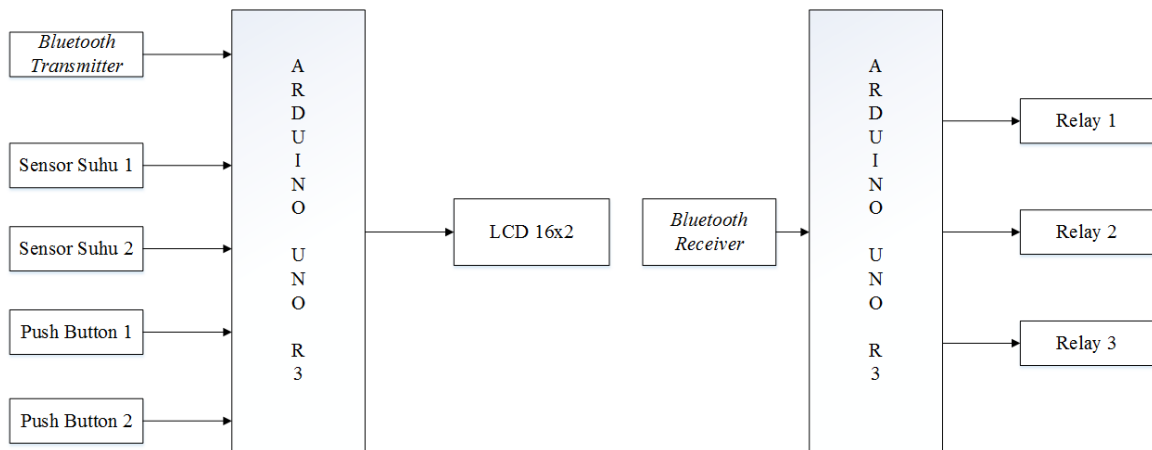
3.3.9 Pembuatan Kesimpulan Dan Saran

Setelah seluruh proses penelitian selesai, dibuat beberapa kesimpulan dari penelitian serta saran untuk penelitian selanjutnya.

3.4 Teknik dan Prosedur Pengumpulan Data

3.4.1 Diagram Blok Alat

Gambar 3.4 merupakan diagram blok dari sistem pengaturan suhu *Air Conditioner* (AC) secara otomatis dimana terdapat 2 buah sensor suhu *thermocouple*, 2 buah *push button* dan modul *bluetooth* HC-05 sebagai masukan (*input*) pada *remote control* pertama. Sensor suhu *thermocouple* digunakan sebagai komponen yang berfungsi mendeteksi nilai suhu udara luar serta nilai suhu udara didalam ruangan dan *push button* digunakan sebagai penghitung (*counter*) jumlah orang yang masuk atau keluar dari ruangan tersebut.



Gambar 3.4 Diagram Blok Alat

(Sumber : dokumentasi pribadi)

Keterangan Gambar :

Sensor Suhu 1 : sensor untuk mendeteksi nilai suhu luar

Sensor Suhu 2 : sensor untuk mendeteksi nilai suhu udara didalam ruangan

Push Button 1 : sebagai penghitung jumlah orang yang masuk ruangan

Push Button 2 : sebagai penghitung jumlah orang yang keluar ruangan

LCD 16x2 : Untuk menampilkan informasi suhu udara diluar, suhu udara didalam ruangan, jumlah penghuni ruangan yang ada didalam ruangan dan nilai *setting* suhu ideal pada AC

Bluetooth Transmitter : Sebagai media pengirim data *serial* dari *Arduino Master* menuju *Arduino Slave*

Bluetooth Receiver : Sebagai media penerima data *serial* dari *Arduino Master* menuju *Arduino Slave*

Relay 1 : Sebagai *automatic switch ON/OFF* yang menjadi pengganti cara menekan tombol secara manual pada *remote AC*

Relay 2 : Sebagai *automatic switch increase temperature* yang menjadi pengganti cara menekan tombol secara manual pada *remote AC*

Relay 3 : Sebagai *automatic switch decrease temperature* yang menjadi pengganti cara menekan tombol secara manual pada *remote AC*

3.4.2 Perancangan Perangkat Keras (*Hardware*)

Perancangan perangkat keras (*hardware*) bertujuan agar sistem memiliki ketepatan dalam menjalankan fungsinya yaitu dapat menentukan nilai *setting* suhu pada *Air Conditioner (AC)* dengan membaca nilai suhu melalui sensor suhu *thermocouple* dan pendeteksian jumlah orang menggunakan *push button* yang kemudian data dari sensor suhu dan *push button* dikirim ke mikrokontroler *Arduino* sebagai masukan (*input*) yang kemudian diolah oleh *Arduino* untuk mendapatkan keluaran (*output*) berupa *setting* nilai suhu AC yang sesuai dengan hasil perhitungan dengan metode logika *fuzzy* yang telah diterapkan didalam kode program yang sesuai dengan kondisi ruangan dan menampilkan keterangan suhu dan nilai *setting* AC pada layar *LCD*.

3.4.2.1 Perancangan dan Pengintegrasian Sistem Arduino

Pada perancangan *prototype* sistem pengaturan suhu *Air Conditioner (AC)* , terdapat dua sensor masukan (*input*) yaitu *push button* dan pembaca nilai suhu melalui sebuah sensor suhu *thermocouple*, unit proses sebuah *Arduino* dengan *chip processor* *ATMega 328* dan keluaran (*output*) berupa *relay* dan *LCD*.

Push button pada sistem ini berfungsi sebagai penghitung (*counter*) penghuni yang masuk ataupun keluar dari dalam ruangan dan sensor suhu *thermocouple* berfungsi sebagai pendeteksi nilai suhu udara baik dari luar ataupun didalam ruangan. Pada bagian *output* terdapat 1 buah *remote control* yang merupakan *output* utama serta LCD yang berfungsi menampilkan informasi suhu, banyaknya jumlah penghuni dan *setting* suhu ideal pada AC.

Sebelum melakukan perancangan alat secara keseluruhan, ada baiknya menyiapkan perancangan integrasi komponen yang akan digunakan dalam alat, guna menentukan *pin input* maupun *output* yang akan digunakan pada mikrokontroler *Arduino Uno R3*. Adapun perancangan tersebut sebagai berikut :

1. Perancangan dan pengintegrasian *push button* dengan *arduino uno R3*
2. Perancangan dan pengintegrasian sensor suhu *thermocouple* dengan *Arduino Uno R3*
3. Perancangan dan pengintegrasian *LCD* dengan *Arduino Uno R3*
4. Perancangan dan pengintegrasian relay dengan *Arduino Uno R3*
5. Perancangan dan pengintegrasian *bluetooth transmitter* dan *bluetooth receiver*

Tabel 3.2 dan tabel 3.3 merupakan tabel pin I/O pada *Arduino* yang digunakan dalam pengintegrasian dengan *input* dan *output*.

Tabel 3.2 Daftar Pin I/O Arduino Dengan Bluetooth Transmitter yang Diintegrasikan Dengan Input Dan Output

Pin I/O Arduino Uno R3	Integrasi
2,3,4,5 dan 6	Sensor Suhu <i>thermocouple 1</i>
8,9,10,11 dan 12	Sensor Suhu <i>thermocouple 2</i>
A0	<i>Push Button 1</i>
A1	<i>Push Button 2</i>
0 dan 1	HC-05 <i>Bluetooth Module</i>
A4 dan A5	LCD

Tabel 3.3 Daftar Pin I/O Arduino Dengan Bluetooth Receiver Yang Diintegrasikan Dengan Input Dan Output

Pin I/O Arduino Uno R3	Integrasi
3,4 dan 5	Relay
0 dan 1	HC-05 <i>Bluetooth Module</i>

3.4.2.2 Perancangan Maket

Pada perancangan sistem perangkat keras (*hardware*) berupa sistem pengaturan suhu *Air Conditioner (AC)*, dibuat dua buah *box* dimana satu *box* berfungsi sebagai sistem utama dan *box* yang lainnya berfungsi sebagai alat *monitoring* dan kontrol. Untuk sistem utama, *remote control* beserta komponen-komponen kontrol lainnya diletakkan didalam sebuah maket berupa sebuah mini *box* berbentuk balok yang memiliki ukuran 20cm x 15cm x 10cm. Untuk sistem *monitoring* dan kontrol, seluruh komponen kontrol juga diletakkan didalam sebuah maket berupa sebuah mini *box* berbentuk balok yang memiliki ukuran 20cm x 15cm x 10cm. Gambar dari alat yang telah dibuat dapat dilihat pada Lampiran 1 dan untuk skema rangkaian dapat dilihat pada Lampiran 2.

3.4.3 Perancangan Perangkat Lunak (*Software*)

Perancangan perangkat lunak berupa perancangan program yang dibuat untuk mendukung sistem kerja dari pengatur kecepatan kipas pendingin untuk mendapatkan sistem yang baik. Adapun perancangan perangkat lunak yang dibuat adalah perancangan program untuk mikrokontroler *Arduino Uno R3* dan pembuatan simulasi perhitungan logika *fuzzy* dengan menggunakan *software* MATLAB.

3.4.3.1 Pembuatan *Rules* Logika *Fuzzy* Menggunakan *Software* MATLAB

Simulasi perhitungan logika *fuzzy* dirancang dengan membuat aturan-aturan (*rules*) *fuzzy* serta melihat gambar permukaan (*surface*) dari perhitungan logika *fuzzy* pada *software* MATLAB. Gambar 3.5 sampai dengan gambar 3.8 merupakan aturan-aturan dan gambar permukaan perhitungan logika *fuzzy* yang dibuat melalui *software* MATLAB.

```

1. If (Luar is Sejuk) and (Dalam is Sejuk) and (Jumlah is Sepi) then (AC is Tinggi) (1)
2. If (Luar is Sejuk) and (Dalam is Sejuk) and (Jumlah is Sedang) then (AC is Tinggi) (1)
3. If (Luar is Sejuk) and (Dalam is Sejuk) and (Jumlah is CukupRamai) then (AC is Tinggi) (1)
4. If (Luar is Sejuk) and (Dalam is Sejuk) and (Jumlah is Padat) then (AC is Tinggi) (1)
5. If (Luar is Sejuk) and (Dalam is Sejuk) and (Jumlah is SangatPadat) then (AC is Tinggi) (1)
6. If (Luar is Normal) and (Dalam is Sejuk) and (Jumlah is Sepi) then (AC is Tinggi) (1)
7. If (Luar is Normal) and (Dalam is Sejuk) and (Jumlah is Sedang) then (AC is Tinggi) (1)
8. If (Luar is Normal) and (Dalam is Sejuk) and (Jumlah is CukupRamai) then (AC is Tinggi) (1)
9. If (Luar is Normal) and (Dalam is Sejuk) and (Jumlah is Padat) then (AC is Sedang) (1)
10. If (Luar is Normal) and (Dalam is Sejuk) and (Jumlah is SangatPadat) then (AC is Sedang) (1)
11. If (Luar is Hangat) and (Dalam is Sejuk) and (Jumlah is Sepi) then (AC is Sedang) (1)
12. If (Luar is Hangat) and (Dalam is Sejuk) and (Jumlah is Sedang) then (AC is Sedang) (1)
13. If (Luar is Hangat) and (Dalam is Sejuk) and (Jumlah is CukupRamai) then (AC is Rendah) (1)
14. If (Luar is Hangat) and (Dalam is Sejuk) and (Jumlah is Padat) then (AC is Rendah) (1)
15. If (Luar is Hangat) and (Dalam is Sejuk) and (Jumlah is SangatPadat) then (AC is Rendah) (1)
16. If (Luar is Sejuk) and (Dalam is Normal) and (Jumlah is Sepi) then (AC is Tinggi) (1)
17. If (Luar is Sejuk) and (Dalam is Normal) and (Jumlah is Sedang) then (AC is Tinggi) (1)

```

Gambar 3.5 *Rules* (1 - 17)

(Sumber : dokumentasi pribadi)

-
18. If (Luar is Sejuk) and (Dalam is Normal) and (Jumlah is CukupRamai) then (AC is Tinggi) (1)
 19. If (Luar is Sejuk) and (Dalam is Normal) and (Jumlah is Padat) then (AC is Sedang) (1)
 20. If (Luar is Sejuk) and (Dalam is Normal) and (Jumlah is SangatPadat) then (AC is Sedang) (1)
 21. If (Luar is Normal) and (Dalam is Normal) and (Jumlah is Sepi) then (AC is Tinggi) (1)
 22. If (Luar is Normal) and (Dalam is Normal) and (Jumlah is Sedang) then (AC is Sedang) (1)
 23. If (Luar is Normal) and (Dalam is Normal) and (Jumlah is CukupRamai) then (AC is Sedang) (1)
 24. If (Luar is Normal) and (Dalam is Normal) and (Jumlah is Padat) then (AC is Rendah) (1)
 25. If (Luar is Normal) and (Dalam is Normal) and (Jumlah is SangatPadat) then (AC is Rendah) (1)
 26. If (Luar is Hangat) and (Dalam is Normal) and (Jumlah is Sepi) then (AC is Sedang) (1)
 27. If (Luar is Hangat) and (Dalam is Normal) and (Jumlah is Sedang) then (AC is Sedang) (1)
 28. If (Luar is Hangat) and (Dalam is Normal) and (Jumlah is CukupRamai) then (AC is Rendah) (1)
 29. If (Luar is Hangat) and (Dalam is Normal) and (Jumlah is Padat) then (AC is Rendah) (1)
 30. If (Luar is Hangat) and (Dalam is Normal) and (Jumlah is SangatPadat) then (AC is Rendah) (1)
 31. If (Luar is Sejuk) and (Dalam is Hangat) and (Jumlah is Sepi) then (AC is Sedang) (1)
 32. If (Luar is Sejuk) and (Dalam is Hangat) and (Jumlah is Sedang) then (AC is Sedang) (1)
 33. If (Luar is Sejuk) and (Dalam is Hangat) and (Jumlah is CukupRamai) then (AC is Sedang) (1)
 34. If (Luar is Sejuk) and (Dalam is Hangat) and (Jumlah is Padat) then (AC is Rendah) (1)
-

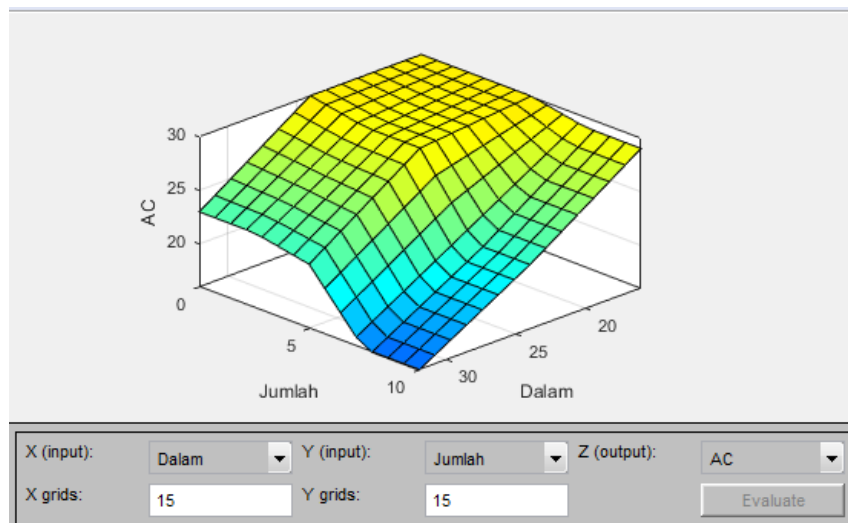
Gambar 3.6 Rules (18 - 34)

(Sumber : dokumentasi pribadi)

35. If (Luar is Sejuk) and (Dalam is Hangat) and (Jumlah is SangatPadat) then (AC is Rendah) (1)
36. If (Luar is Normal) and (Dalam is Hangat) and (Jumlah is Sepi) then (AC is Sedang) (1)
37. If (Luar is Normal) and (Dalam is Hangat) and (Jumlah is Sedang) then (AC is Sedang) (1)
38. If (Luar is Normal) and (Dalam is Hangat) and (Jumlah is CukupRamai) then (AC is Rendah) (1)
39. If (Luar is Normal) and (Dalam is Hangat) and (Jumlah is Padat) then (AC is Rendah) (1)
40. If (Luar is Normal) and (Dalam is Hangat) and (Jumlah is SangatPadat) then (AC is Rendah) (1)
41. If (Luar is Hangat) and (Dalam is Hangat) and (Jumlah is Sepi) then (AC is Sedang) (1)
42. If (Luar is Hangat) and (Dalam is Hangat) and (Jumlah is Sedang) then (AC is Rendah) (1)
43. If (Luar is Hangat) and (Dalam is Hangat) and (Jumlah is CukupRamai) then (AC is Rendah) (1)
44. If (Luar is Hangat) and (Dalam is Hangat) and (Jumlah is Padat) then (AC is Rendah) (1)
45. If (Luar is Hangat) and (Dalam is Hangat) and (Jumlah is SangatPadat) then (AC is Rendah) (1)

Gambar 3.7 Rules (35 - 45)

(Sumber : dokumentasi pribadi)



Gambar 3.8 Surface

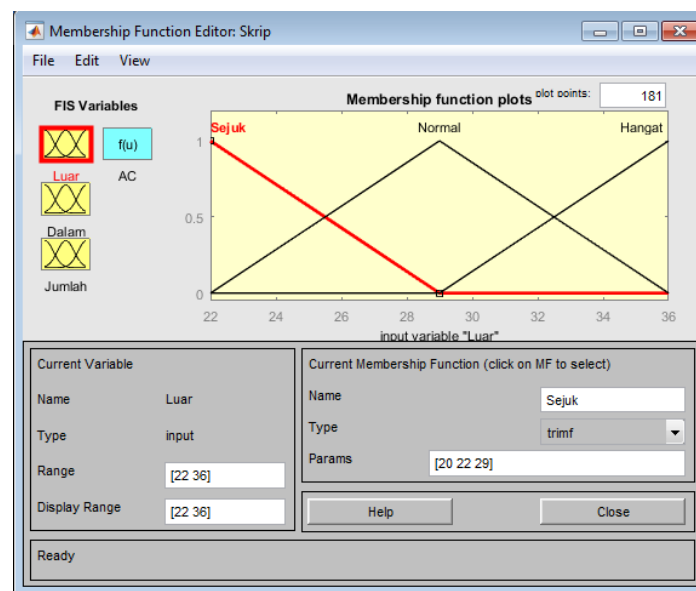
(Sumber : dokumentasi pribadi)

3.4.3.2 Perancangan Program *Arduino* Menggunakan *software IDE Arduino 1.8.0*

Perancangan program dibuat berdasarkan prinsip kerja dari sistem pengaturan *setting* nilai suhu ideal pada *Air Conditioner* (AC) berdasarkan jumlah penghuni dan suhu udara luar dengan penerapan metode kontrol logika *fuzzy* berbasis *Arduino Uno R3*. Program logika *fuzzy* dibuat berdasarkan teori perhitungan logika *fuzzy* dimana perhitungan logika *fuzzy* dibagi menjadi 3 tahap, yaitu : fuzzyfikasi, inferensi dan defuzzyfikasi.

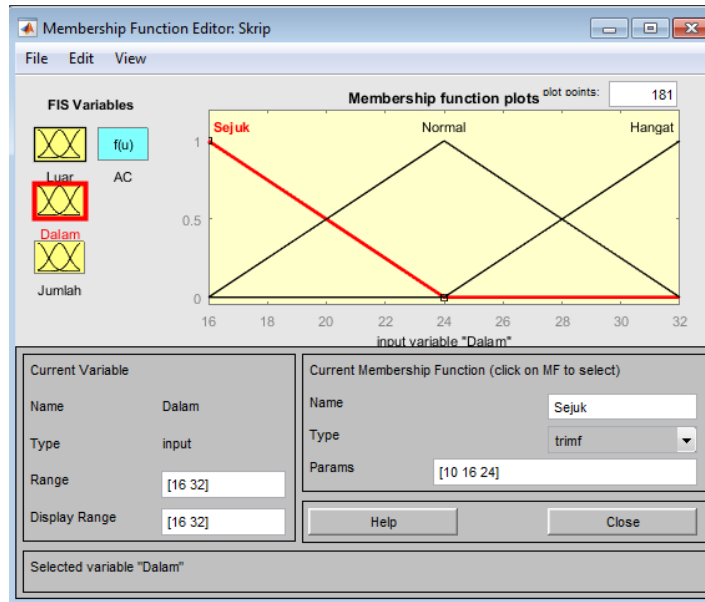
1. Fuzzyfikasi

Pada tahap fuzzyfikasi, nilai *crisp* dari *input* suhu udara luar, suhu udara didalam ruangan dan jumlah penghuni ruangan serta nilai *crisp* dari *output setting* AC dipetakan kedalam variabel linguistik. Bentuk diagram fungsi keanggotaan variabel linguistik *input* suhu udara luar, suhu udara didalam ruangan, jumlah penghuni ruangan dan *output setting* AC dapat dilihat pada gambar 3.9 sampai dengan 3.12 berikut.

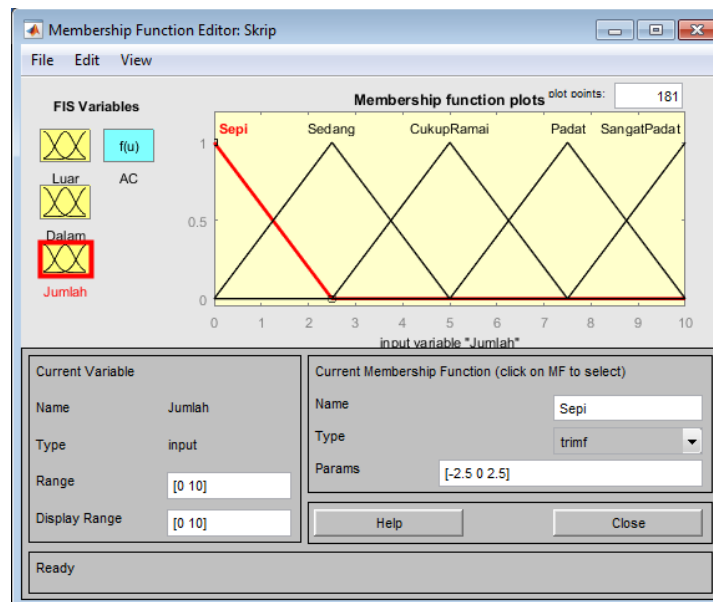


Gambar 3.9 Fungsi Keanggotaan Suhu Luar

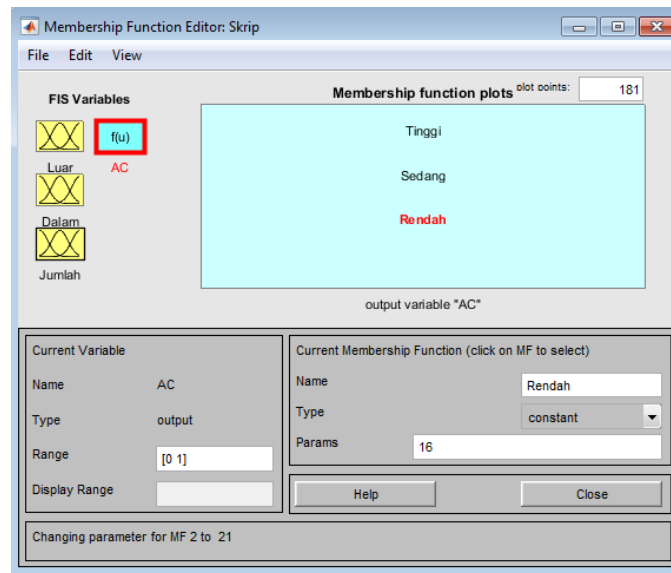
(Sumber : dokumentasi pribadi)



Gambar 3.10 Fungsi Keanggotaan Suhu Dalam
(Sumber : dokumentasi pribadi)



Gambar 3.11 Fungsi Keanggotaan Jumlah Penghuni
(Sumber : dokumentasi pribadi)



Gambar 3.12 Fungsi Keanggotaan Setting AC

(Sumber : dokumentasi pribadi)

2. Inferensi

Tahap kedua adalah inferensi, yaitu tahap pembuatan aturan (*rules*) *fuzzy*. Rules yang dibuat dapat dilihat pada gambar 3.5 sampai dengan gambar 3.7 pada poin sebelumnya. Pada proses inferensi, metode yang digunakan adalah metode sugeno.

3. Defuzzyfikasi

Tahap terakhir adalah tahap defuzzyfikasi, yaitu tahap perubahan nilai *fuzzy* menjadi nilai *crisp* kembali dengan metode *weighted-average* dengan rumus.

$$y^* = \sum \frac{\mu(y)y}{\mu(y)} \quad (3.1)$$

di mana y adalah nilai *crisp* dan $\mu(y)$ adalah derajat keanggotaan dari nilai *crisp* y .

3.4.4 Pengujian Alat dan Pengambilan Data

Prosedur penelitian dan pengumpulan data pada sistem pengaturan suhu pada *Air Conditioner* (AC) berdasarkan suhu udara luar, suhu udara didalam ruangan dan

jumlah penghuni ruangan dengan penerapan metode kontrol logika *fuzzy* berbasis *Arduino Uno R3* dilakukan melalui beberapa tahap:

1. Penelitian terkait pengaruh perubahan kondisi suhu luar dan jumlah penghuni didalam suatu ruangan terhadap nilai suhu didalam ruangan

Peneliti melakukan analisa terkait pengaruh perubahan kondisi suhu luar dan jumlah penghuni didalam suatu ruangan terhadap perubahan nilai suhu didalam suatu ruangan untuk mengetahui apakah suhu udara diluar ruangan dan jumlah penghuni yang berada didalam ruangan akan mempengaruhi besarnya nilai suhu yang terukur didalam ruangan tersebut.

2. Pembuatan rangkaian sistem pengaturan suhu pada *Air Conditioner* (AC)

Desain sistem pengaturan suhu pada *Air Conditioner* (AC) dirancang dengan menggunakan *software Google SketchUp 2017* dalam bentuk desain 3D. Rangkaian sistem pengaturan *setting* suhu pada *Air Conditioner* (AC) menggunakan mikrokontroler *Arduino Uno R3* yang telah diintegrasikan dengan komponen-komponen pendukung seperti : *push button*, sensor suhu *thermocouple*, LCD, relay dan modul *bluetooth HC-05* dengan menggunakan wadah berupa dua buah kotak hitam (*black box*).

3. Pembuatan program mikrokontroler ATmega 328 dengan *Arduino IDE 1.8.0*

Program mikrokontroler ATmega 328 dibuat untuk memproses *input* data dari *push button* dan sensor suhu *thermocouple*, melakukan perhitungan logika *fuzzy* ketika pemrosesan data di mikrokontroler, menampilkan nilai suhu pada LCD dan memberikan *output* berupa nilai *setting* suhu ideal pada AC sehingga nilai suhu pada

AC dapat di-*set* sesuai dengan suhu ideal.. Program dibuat dengan bahasa *Arduino* dan menggunakan *software Arduino IDE*.

4. Pengujian pengaruh kondisi suhu udara luar terhadap kondisi suhu udara didalam ruangan

Pengujian pengaruh kondisi suhu udara luar terhadap kondisi suhu udara didalam ruangan dilakukan untuk mengetahui seberapa besar pengaruh kondisi suhu udara luar terhadap suhu udara didalam ruangan.

5. Pengujian pengaruh jumlah penghuni ruangan terhadap kondisi suhu udara didalam ruangan

Pengujian pengaruh jumlah penghuni ruangan terhadap kondisi suhu udara didalam ruangan dilakukan untuk mengetahui seberapa besar pengaruh jumlah penghuni ruangan terhadap suhu udara didalam ruangan.

6. Pengujian pembacaan nilai suhu pada sensor suhu *thermocouple*

Pengujian pengukuran tegangan pada sensor suhu *thermocouple* bertujuan untuk mengetahui kondisi dan kinerja dari komponen tersebut. Alat ukur atau instrumen yang digunakan adalah berupa *thermometer digital*. Kemudian hasil pengukuran pada sensor dibandingkan dengan hasil pembacaan pada *thermometer*.

7. Pengujian *push button*

Pengujian *push button* dilakukan untuk mengetahui apakah *push button* dapat bekerja dengan baik ketika ditekan.

8. Pengujian jarak komunikasi secara *wireless* antara dua modul *bluetooth HC-05*

Pengujian jarak komunikasi dua modul *bluetooth HC-05* dilakukan dengan menghubungkan kedua modul *bluetooth HC-05* dan perlahan memperjauh jarak antara

kedua modul tersebut hingga koneksi antara kedua modul *bluetooth* HC-05 tersebut terputus.

9. Pengujian *input* suhu dan jumlah penghuni terhadap nilai *setting* AC

3.4.4.1 Pengujian pengaruh kondisi suhu udara luar terhadap nilai suhu didalam ruangan

Pengujian pengaruh kondisi suhu udara luar terhadap nilai suhu didalam ruangan dilakukan dengan memanfaatkan kenaikan suhu udara luar pada siang hari menjelang sore hari. Lalu dilakukan pengukuran suhu udara didalam ruangan untuk mengetahui apakah terdapat pengaruh dari kenaikan suhu udara diluar terhadap nilai suhu udara yang ada didalam ruangan. Tabel hasil pengamatan dapat dilihat pada **tabel**

3.4.

Tabel 3.4 Hasil pengujian kondisi suhu udara luar terhadap suhu udara didalam ruangan

No	Waktu	Nilai Suhu Luar	Nilai Suhu Didalam Ruangan
1			
2			
3			
4			
5			
6			
7			
8			

3.4.4.2 Pengujian pengaruh jumlah penghuni ruangan terhadap nilai suhu didalam ruangan

Pengujian pengaruh jumlah penghuni ruangan terhadap nilai suhu didalam ruangan dilakukan dengan memasukkan beberapa orang kedalam ruangan tersebut lalu melihat perubahan nilai suhu didalam ruangan tersebut setiap 10 menit. Kemudian, mengeluarkan orang tersebut satu per satu untuk kembali melihat perubahan nilai suhu yang terjadi didalam ruangan tersebut. Data hasil pengamatan pengaruh jumlah penghuni ruangan terhadap nilai suhu didalam ruangan dapat dilihat pada **tabel 3.5** berikut ini.

Tabel 3.5 Data hasil pengamatan pengujian jumlah penghuni ruangan terhadap suhu udara didalam ruangan

No	Jumlah Penghuni	Nilai Suhu Didalam Ruangan (Setelah 10 menit)
1		
2		
3		
4		
5		
6		
7		

3.4.4.3 Pengujian dan pengukuran pembacaan sensor suhu *thermocouple* sebagai pendeteksi suhu udara luar

Pengujian sensor dilakukan pada sensor suhu dengan jenis *thermocouple*. *Output* sensor dihubungkan ke *pin* nomor 2,3,4,5 dan 6 pada *Arduino*. Tegangan

sumber yang digunakan untuk mengaktifkan sensor adalah tegangan 5 volt DC dari *Arduino*. Pengujian sensor suhu *thermocouple* dilakukan dengan memasukkan sensor kedalam air hangat dan melihat kenaikan suhu yang dibaca oleh sensor *thermocouple* dan membandingkan dengan pembacaan dari *thermometer digital* untuk mengetahui seberapa besar nilai kesalahan (*error*) dari pembacaan sensor *thermocouple*. **Tabel 3.6** merupakan data hasil pengamatan pengujian nilai tegangan dan pembacaan logika sensor ketika terjadi kenaikan nilai suhu. Rumus untuk menghitung persentase kesalahan dari pembacaan sensor *thermocouple* dapat dilihat pada **Lampiran 4**.

Tabel 3.6 Data hasil pengamatan pengujian nilai tegangan dan pembacaan sensor suhu luar

No	Waktu	Nilai Suhu Pada <i>Thermocouple</i> 1	Nilai Suhu Pada <i>Thermometer</i>	Persentase Kesalahan (%)
1	0			
2	5			
3	10			
4	15			
5	20			
6	25			
7	30			

3.4.4.4 Pengujian dan pengukuran pembacaan sensor suhu *thermocouple* sebagai pendeteksi suhu udara dalam ruangan

Pengujian sensor dilakukan pada sensor suhu dengan jenis *thermocouple*. *Output* sensor dihubungkan ke *pin* nomor 7,8,9,10 dan 11 pada *Arduino*. Tegangan sumber

yang digunakan untuk mengaktifkan sensor adalah tegangan 5 volt DC dari *Arduino*. Pengujian sensor suhu *thermocouple* dilakukan dengan memasukkan sensor kedalam air hangat dan melihat kenaikan suhu yang dibaca oleh sensor *thermocouple* dan membandingkan dengan pembacaan dari *thermometer digital* untuk mengetahui seberapa besar nilai kesalahan (*error*) dari pembacaan sensor *thermocouple*. **Tabel 3.7** merupakan data hasil pengamatan pengujian nilai tegangan dan pembacaan logika sensor ketika terjadi kenaikan nilai suhu. Rumus untuk menghitung persentase kesalahan dari pembacaan sensor *thermocouple* dapat dilihat pada **Lampiran 4**.

Tabel 3.7 Data hasil pengamatan pengujian nilai tegangan dan pembacaan sensor suhu didalam ruangan

No	Waktu	Nilai Suhu Pada Pada <i>Thermocouple 2</i>	Nilai Suhu Pada <i>Thermometer</i>	Persentase Kesalahan (%)
1	0			
2	5			
3	10			
4	15			
5	20			
6	25			
7	30			

3.4.4.5 Pengujian *Input Push Button*

Pengujian *push button* dilakukan untuk mengetahui apakah *push button* dapat bekerja dengan baik ketika ditekan. Pengujian dilakukan dengan mengukur tegangan

yang melewati *push button* dan melihat logika yang masuk kedalam kontroler melalui *serial monitor* pada aplikasi *Arduino IDE*. **Tabel 3.8** dan **tabel 3.9** merupakan data hasil pengamatan dari pengujian *input push button*.

Tabel 3.8 Data hasil pengamatan pengujian *input push button* (*pull down resistor*)

No	Kondisi <i>Push Button</i>	Tegangan	Indikator
1	Ditekan		
2	Tidak Ditekan		

Tabel 3.9 Data hasil pengamatan pengujian *input push button* (*pull up resistor*)

No	Kondisi <i>Push Button</i>	Tegangan	Indikator
1	Ditekan		
2	Tidak Ditekan		

3.4.4.6 Pengujian Jarak Pengiriman dan Penerimaan Data Modul *Bluetooth HC-05*

Pengujian jarak pengiriman dan penerimaan data pada modul *bluetooth HC-05* dilakukan untuk mengetahui seberapa jauh jarak yang dapat dijangkau oleh modul *Bluetooth HC-05* untuk dapat mengirim atau menerima data sehingga dapat menyesuaikan jarak antara *device* yang sedang terhubung dengan *bluetooth* agar koneksi tidak terputus. **Tabel 3.10** merupakan tabel hasil pengujian jarak pengiriman dan penerimaan data pada modul *bluetooth HC-05*.

Tabel 3.10 Data hasil pengujian jarak pengiriman dan penerimaan data pada modul *bluetooth* HC-05

No	Jarak	Koneksi
1		
2		
3		
4		
5		
6		

3.4.4.7 Pengujian Input Suhu dan Jumlah Penghuni Terhadap Nilai *Setting* AC

Pengujian *input* suhu terhadap nilai *setting* suhu pada *Air Conditioner* (AC) dilakukan dengan tujuan untuk menyesuaikan nilai suhu dan jumlah penghuni dengan *setting* suhu pada *Air Conditioner* (AC) serta menguji penggunaan metode kontrol logika *fuzzy* yang diterapkan kepada suatu mikrokontroler dan membandingkan dengan perhitungan teori dari logika *fuzzy* untuk mendapatkan suatu kesimpulan tentang penerapan logika *fuzzy* kedalam suatu sistem kontrol.

Nilai *input* suhu yang diberikan dan nilai *setting* suhu AC pada *Arduino* akan dibandingkan dengan hasil perhitungan menggunakan logika *fuzzy* peneliti menggunakan *software* MATLAB dan perhitungan manual. **Tabel 3.11** merupakan data hasil pengamatan pengujian input suhu dan jumlah penghuni terhadap nilai *setting* AC.

Tabel 3.11 Data hasil pengamatan pengujian input suhu dan jumlah penghuni terhadap nilai *setting* AC

No	Nilai Suhu Luar	Nilai Suhu Dalam Ruangan	Jumlah Penghuni	Nilai <i>Setting</i> AC	Nilai Suhu (Setelah 5 Menit)
1					
2					
3					
4					
5					

3.5 Teknik Analisis Data

Pengujian pengaruh kondisi suhu udara luar dan jumlah penghuni terhadap suhu didalam ruangan dilakukan diruang sidang lantai 1 gedung Teknik Elektro Universitas Negeri Jakarta. Untuk pengujian pengaruh suhu luar terhadap suhu didalam ruangan dilakukan pada siang hari menjelang sore hari saat kondisi suhu udara luar mulai meningkat. Lalu ukur suhu udara didalam ruangan dan catat hasilnya. Pada pengujian pengaruh jumlah penghuni terhadap nilai suhu udara didalam ruangan dilakukan dengan memasukkan beberapa orang kedalam ruangan lalu mencatat suhu didalam ruangan tersebut, lalu mengeluarkannya satu per satu dan catat kembali nilai suhu ketika jumlah penghuni mulai berkurang.

Pada pengujian sistem, sistem diaktifkan untuk membuat setiap komponen bekerja lalu dilakukan pengujian pada setiap komponen dan mencatat hasilnya. Pada pengujian sensor suhu *thermocouple* dilakukan pengujian dengan membandingkan hasil pembacaan pada *thermocouple* dan hasil pembacaan pada *thermometer digital*

untuk melihat selisih nilai suhu yang terbaca lalu mencatat hasilnya dan melakukan kalibrasi pada *thermocouple* melalui program pada *Arduino* agar nilai *error thermocouple* menjadi lebih kecil. Untuk pengujian *push button*, dilakukan dengan cara mengukur tegangan yang mengalir pada *push button* dan melihat nilai logika yang terbaca pada *serial monitor* lalu mencatat hasilnya.

Kemudian melakukan pengujian terhadap seluruh sistem untuk mengetahui apakah sistem sudah dapat berjalan dengan baik atau belum. Saat terjadi perubahan nilai suhu dan jumlah penghuni, ukur suhu didalam ruangan setiap 5 menit setelah terjadi perubahan nilai *setting* suhu lalu mencatat hasilnya.

BAB IV

HASIL PENELITIAN

4.1 Deskripsi Hasil Penelitian

Berdasarkan hasil penelitian, didapatkan sebuah sistem pengaturan kecepatan kipas pendingin ruangan otomatis dengan penerapan metode kontrol logika *fuzzy*. Pada pengujian yang telah dilakukan sebelumnya, didapatkan data-data hasil pengukuran sebagai berikut :

4.1.1 Hasil Pengujian Pengaruh Kondisi Suhu Udara Luar terhadap Nilai Suhu Udara Didalam Ruangan

Pengujian pengaruh kondisi suhu udara luar terhadap nilai suhu udara didalam ruangan dilakukan pada siang menjelang sore hari dan perubahan nilai suhu udara dicatat setiap 15 menit. Suhu udara luar yang diukur adalah suhu udara disekitar koridor didepan ruang sidang lantai 1 gedung teknik elektro dan ruangan yang digunakan untuk pengukuran suhu dalam adalah ruang sidang lantai 1 gedung teknik elektro. Data hasil pengamatan pengaruh kondisi suhu udara luar terhadap nilai suhu udara didalam ruangan dapat dilihat pada **tabel 4.1**.

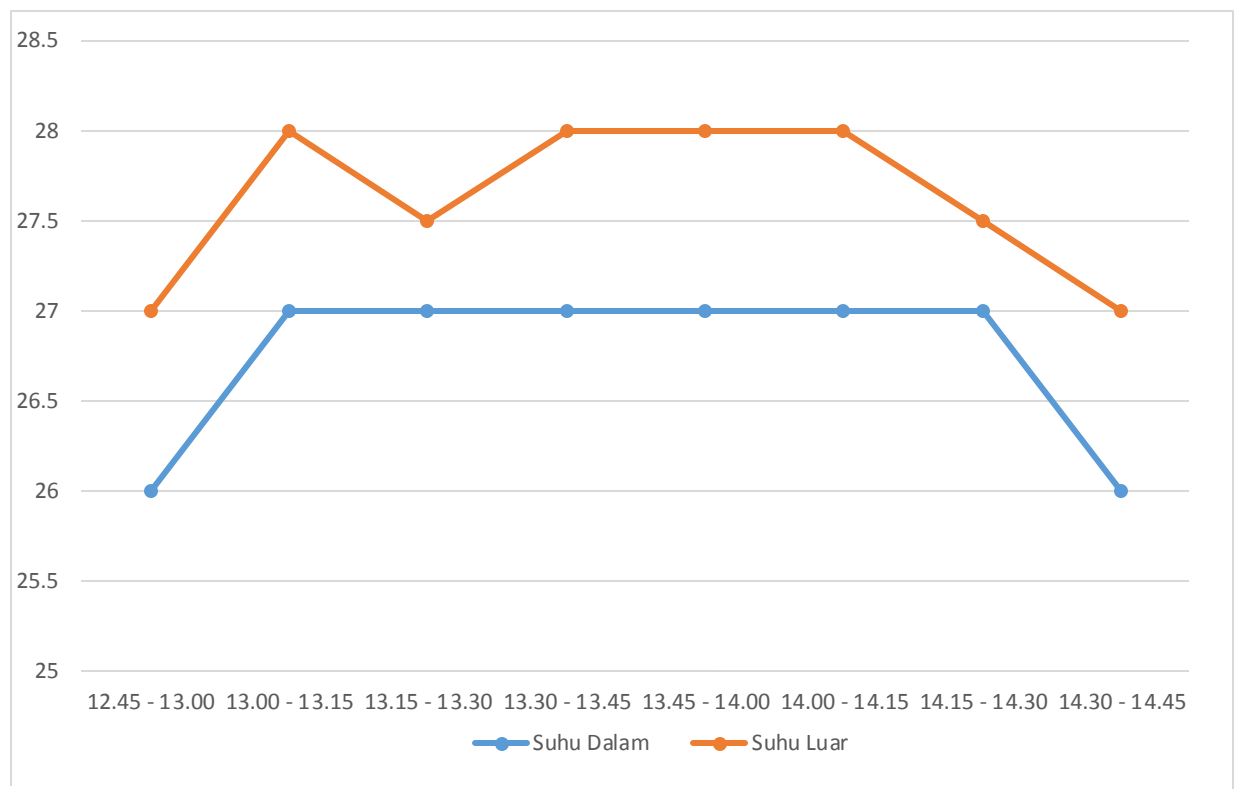
Tabel 4.1 Data hasil pengamatan pengaruh kondisi suhu udara luar terhadap nilai suhu didalam ruangan

No	Waktu	Nilai Suhu Luar	Nilai Suhu Didalam Ruangan
1	12.45 – 13.00	27 °C	26°C
2	13.00 – 13.15	28 °C	27 °C
3	13.15 – 13.30	27,5 °C	27 °C

Tabel 4.1 (Lanjutan)

No	Waktu	Nilai Suhu Luar	Nilai Suhu Didalam Ruangan
4	13.30 – 13.45	28 °C	27 °C
5	13.45 – 14.00	28 °C	27 °C
6	14.00 – 14.15	28 °C	27 °C
7	14.15 – 14.30	27,5 °C	27 °C
8	14.30 – 14.45	27 °C	26 °C

Dari tabel pengujian pengaruh kondisi suhu udara luar terhadap nilai suhu didalam ruangan, didapatkan sebuah grafik seperti pada **gambar 4.1** berikut.



Gambar 4.1 Grafik Pengaruh Suhu Luar Dengan Suhu Dalam

(Sumber : hasil pengujian)

Berdasarkan grafik diatas, kondisi suhu udara didalam ruangan dipengaruhi oleh kondisi suhu udara diluar ruangan. Ketika nilai suhu diluar ruangan naik, maka suhu didalam ruangan juga akan naik. Dan begitupun sebaliknya, ketika nilai suhu diluar ruangan turun, maka suhu didalam ruangan perlahan akan ikut menurun.

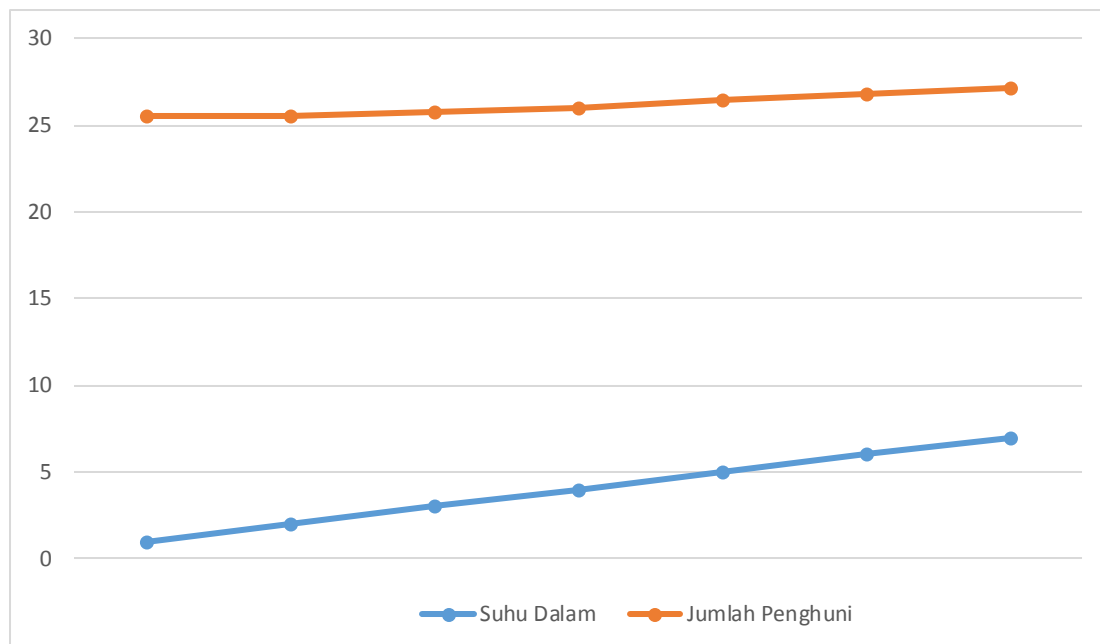
4.1.2 Hasil Pengujian Pengaruh Jumlah Penghuni Ruangan terhadap Nilai Suhu Udara Didalam Ruangan

Pengujian pengaruh jumlah penghuni ruangan terhadap nilai suhu udara didalam ruangan dilakukan dengan memasukkan beberapa orang kedalam ruangan dan mengeluarkannya satu persatu. Data hasil pengujian pengaruh jumlah penghuni ruangan terhadap nilai suhu udara didalam ruangan dapat dilihat pada **tabel 4.2**.

Tabel 4.2 Data hasil pengamatan pengaruh jumlah penghuni ruangan terhadap nilai suhu didalam ruangan

No	Jumlah Penghuni	Nilai Suhu Didalam Ruangan (Setelah 10 Menit)
1	1	25,5 °C
2	2	25,5 °C
3	3	25,8 °C
4	4	26 °C
5	5	26,5 °C
6	6	26,8 °C
7	7	27,2 °C

Dari tabel hasil pengujian pengaruh jumlah penghuni ruangan terhadap nilai suhu didalam ruangan didapatkan sebuah grafik seperti **gambar 4.2** berikut.



Gambar 4.2 Grafik Pengaruh Jumlah Penghuni Ruang Terhadap Nilai Suhu Udara Dalam

(Sumber : hasil pengujian)

Berdasarkan grafik diatas, kondisi suhu udara didalam ruangan dipengaruhi oleh jumlah penghuni yang berada didalam ruangan. Ketika jumlah penghuni ruangan bertambah, maka suhu didalam ruangan akan meningkat. Dan begitupun sebaliknya, ketika jumlah penghuni didalam ruangan berkurang, maka suhu didalam ruangan perlahan akan menurun.

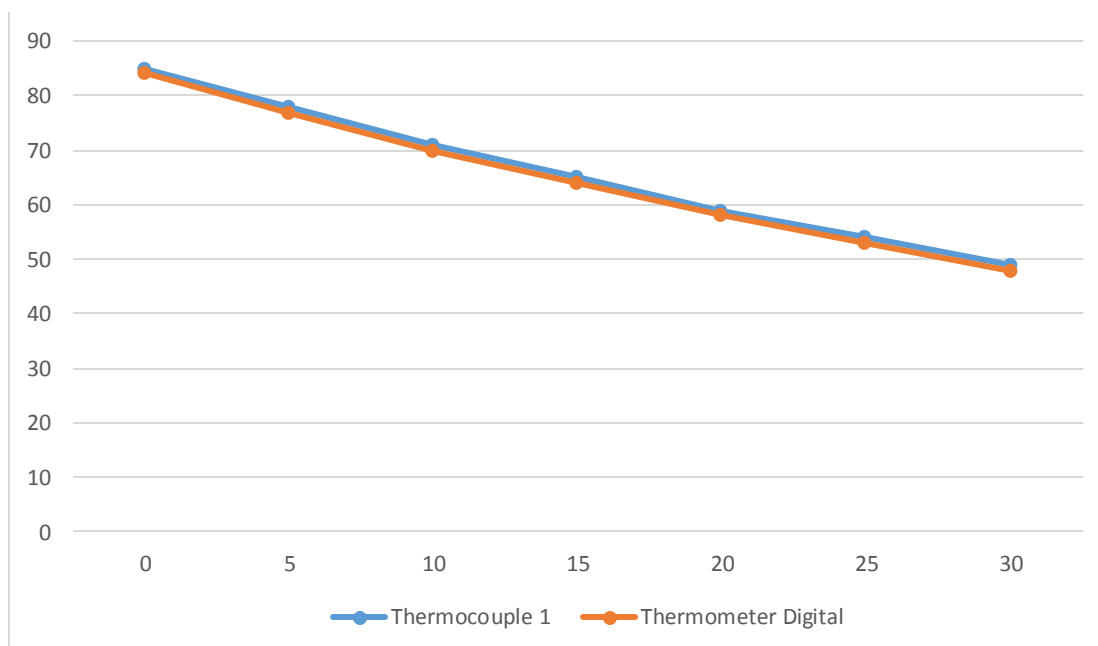
4.1.3 Hasil Pengujian Sensor *Thermocouple* 1 untuk pendeteksi suhu udara luar

Sensor *thermocouple* 1 diuji dengan cara membandingkan hasil pembacaan sensor dan pembacaan thermometer digital menggunakan media berupa air panas dalam waktu 30 menit dan hasilnya dicatat setiap 5 menit. Data hasil pengujian sensor *thermocouple* 1 melalui perbandingan dengan *thermometer digital* dapat dilihat pada **tabel 4.3**.

Tabel 4.3 Data hasil pengujian pembacaan sensor *thermocouple* 1

No	Waktu (Menit)	Nilai Suhu Pada <i>Thermocouple</i> 1	Nilai Suhu Pada <i>Thermometer</i>	Persentase Kesalahan (%)
1	0	85 °C	84 °C	1,2 %
2	5	78 °C	77 °C	1,3 %
3	10	71 °C	70 °C	1,4 %
4	15	65 °C	64 °C	1,5 %
5	20	59 °C	58 °C	1,7 %
6	25	54 °C	53 °C	1,9 %
7	30	49 °C	48 °C	2,0 %

Berdasarkan tabel diatas, didapatkan sebuah grafik seperti **gambar 4.3** berikut.



Gambar 4.3 Grafik Perbandingan Antara *Thermocouple* 1 dan *Thermometer Digital*
(Sumber : hasil pengujian)

Berdasarkan grafik diatas, dapat disimpulkan bahwa nilai suhu pada *thermocouple* 1 memiliki selisih dengan *thermometer digital* dengan rata-rata nilai selisih sebesar 1 °C.

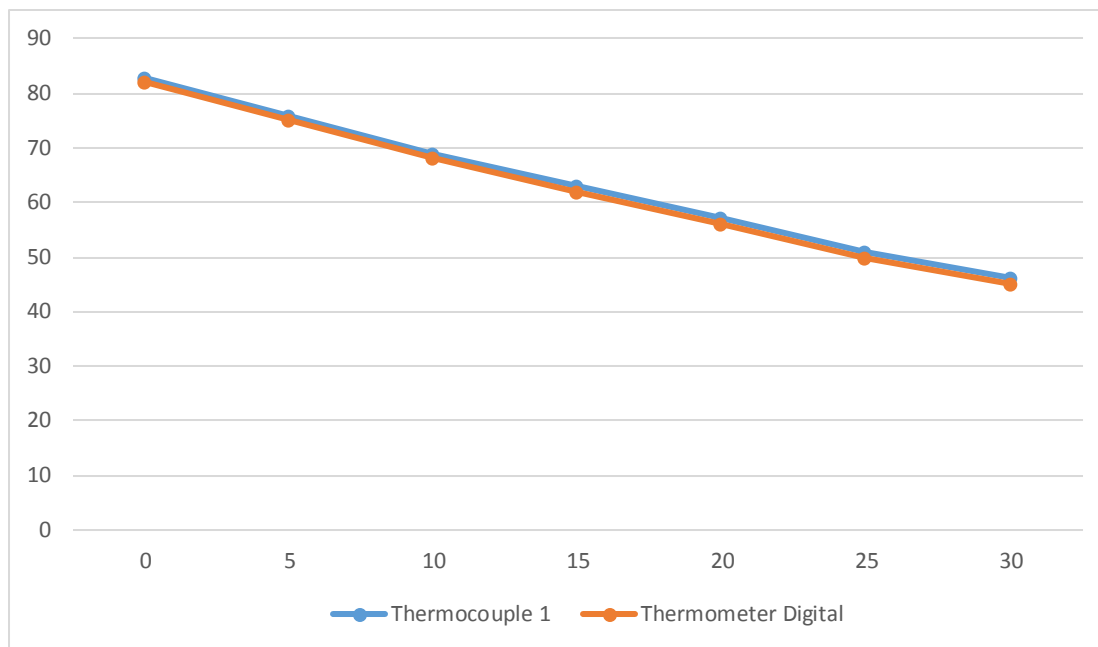
4.1.4 Hasil Pengujian Sensor *Thermocouple* 2 untuk pendeteksi suhu udara didalam ruangan

Sensor *thermocouple* 2 diuji dengan cara membandingkan hasil pembacaan sensor dan pembacaan *thermometer digital* menggunakan media berupa air panas dalam waktu 30 menit dan hasilnya dicatat setiap 5 menit. Pengujian sensor dilakukan dirumah peneliti Data hasil pengujian sensor *thermocouple* 2 melalui perbandingan dengan *thermometer digital* dapat dilihat pada **tabel 4.4**.

Tabel 4.4 Data hasil pengamatan pengujian pembacaan sensor *thermocouple* 2

No	Waktu	Nilai Suhu Pada <i>Thermocouple</i> 2	Nilai Suhu Pada <i>Thermometer</i>	Persentase Kesalahan (%)
1	0	83 °C	82 °C	1,2 %
2	5	76 °C	75 °C	1,3 %
3	10	69 °C	68 °C	1,4 %
4	15	63 °C	62 °C	1,6 %
5	20	57 °C	56 °C	1,8 %
6	25	51 °C	50 °C	2,0 %
7	30	46 °C	45 °C	2,2 %

Berdasarkan tabel diatas, didapatkan sebuah grafik seperti **gambar 4.2** berikut.



Gambar 4.4 Grafik Perbandingan Antara *Thermocouple 2* dan *Thermometer Digital*
(Sumber : hasil pengujian)

Berdasarkan grafik diatas, dapat disimpulkan bahwa nilai suhu pada *thermocouple 2* memiliki selisih dengan *thermometer digital* dengan rata-rata nilai selisih sebesar 1 °C.

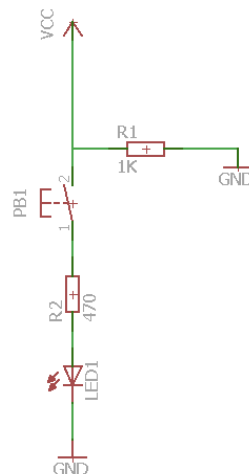
4.1.5 Hasil Pengujian *Input Push Button*

Pengujian *push button* dilakukan dengan cara mengukur nilai tegangan pada *push button* dan melihat nilai logika dari *push button* tersebut saat *push button* berada pada kondisi ditekan dan tidak ditekan. Data hasil pengujian *input push button* dapat dilihat pada **tabel 4.5** dan **tabel 4.6**.

Tabel 4.5 Data hasil pengamatan pengujian *input push button (pull down resistor)*

No	Kondisi <i>Push Button</i>	Tegangan	Indikator
1	Ditekan	4,8 volt	<i>ON</i>
2	Tidak Ditekan	4,8 volt	<i>OFF</i>

Gambar rangkaian pengujian *input push button* dengan *pull up resistor* dapat dilihat pada **gambar 4.5** berikut.

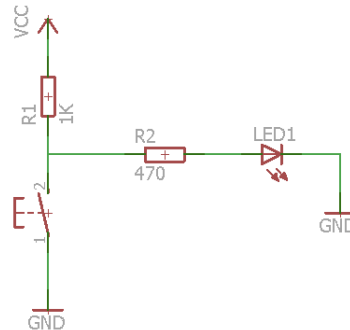


Gambar 4.5 Skema *Push Button* Dengan *Pull Down Resistor*
(Sumber : Dokumentasi Pribadi)

Tabel 4.6 Data hasil pengamatan pengujian *input push button (pull up resistor)*

No	Kondisi <i>Push Button</i>	Tegangan	Indikator
1	Ditekan	4,8 volt	<i>OFF</i>
2	Tidak Ditekan	4,8 volt	<i>ON</i>

Gambar rangkaian pengujian *input push button* dengan *pull upresistor* dapat dilihat pada **gambar 4.6** berikut.



Gambar 4.6 Skema *Push Button* Dengan *Pull Up Resistor*
(Sumber : dokumentasi pribadi)

4.1.6 Hasil Pengujian jarak pengiriman dan penerimaan data pada modul *bluetooth* HC-05

Pengujian jarak pengiriman dan penerimaan data pada modul *bluetooth* HC-05 dilakukan dengan menghubungkan modul *bluetooth* HC-05 dengan perangkat modul *bluetooth* HC-05 lainnya lalu perlahan memperjauh jarak antara kedua modul *bluetooth* HC-05 hingga koneksi terputus. Data hasil pengujian jarak pengiriman dan penerimaan data pada modul *bluetooth* HC-05 dapat dilihat pada **tabel 4.7** berikut.

Tabel 4.7 Pengujian jarak pengiriman dan penerimaan data pada modul *bluetooth* HC-05

No	Jarak	Koneksi
1	1 meter	Terhubung
2	2 meter	Terhubung
3	5 meter	Terhubung
4	7 meter	Terhubung
5	9 meter	Terhubung
6	11 meter	Terputus

4.1.7 Hasil Pengujian *Input* Suhu dan Jumlah Penghuni Ruangan Terhadap Nilai *setting* AC

Data hasil pengujian *input* nilai suhu dan jumlah penghuni ruangan terhadap nilai *setting* AC dapat dilihat pada **tabel 4.8**.

Tabel 4.8 Data hasil Pengujian *Input* Suhu dan Jumlah Penghuni Ruangan Terhadap Nilai *Setting* AC

No	Nilai Suhu Luar	Nilai Suhu Dalam Ruangan	Jumlah Penghuni	Nilai <i>setting</i> AC	Nilai Suhu (Setelah 10 Menit)
1	28 °C	25 °C	1	24 °C	24°C
2	28 °C	23 °C	2	23 °C	23 °C
3	27 °C	23 °C	3	23 °C	23 °C
4	26 °C	23 °C	4	23 °C	23 °C
5	26 °C	23 °C	5	23°C	23 °C

4.1.8 Cara Kerja Alat

Ketika sistem pada *remote control* diaktifkan, maka *remote control* akan mulai melakukan perhitungan untuk menentukan nilai *setting* pada AC (*Air Conditioner*). Ketika tombol *counter up* ditekan, maka nilai jumlah penghuni ruangan akan bertambah. Lalu *remote control* akan melakukan perhitungan kembali dengan menyesuaikan nilai input yang berubah lalu merubah nilai *setting* pada AC (*Air Conditioner*) sesuai dengan hasil perhitungan. Ketika tombol *counter down* ditekan, maka nilai jumlah penghuni ruangan akan berkurang. Lalu *remote control* akan melakukan perhitungan kembali dengan menyesuaikan nilai input yang berubah lalu

merubah nilai *setting* pada AC (*Air Conditioner*) sesuai dengan hasil perhitungan. Ketika terjadi perubahan nilai suhu yang dideteksi oleh sensor, *remote control* juga akan melakukan perhitungan dengan menyesuaikan nilai input yang berubah lalu merubah nilai *setting* pada AC (*Air Conditioner*) sesuai dengan hasil perhitungan.

4.2 Analisis Data Penelitian

4.2.1 Analisis Sensor Suhu

Sensor suhu yang digunakan adalah sensor suhu dengan jenis *thermocouple type K* dengan menggunakan *driver module* MAX6675 sebagai jembatan untuk pengiriman data dari sensor ke *Arduino*. Nilai pembacaan suhu pada *thermocouple type K* bergantung pada pemasangan *probe thermocouple* dengan *driver module* MAX6675. Apabila pemasangan *probe thermocouple* dengan *driver module* MAX6675 terbalik, maka pembacaan nilai suhu juga akan terbalik.

4.2.2 Analisis Push Button

Push button digunakan dengan posisi *pulldown switch* dimana *push button* memiliki nilai tegangan sebesar 4,8 volt dan ketika *push button* berada pada posisi tidak ditekan, led indikator menyala dan ketika *push button* berada pada posisi ditekan, led indikator mati. Selanjutnya *push button* digunakan dengan posisi *pullup switch* dimana *push button* memiliki nilai tegangan sebesar 4,8 volt dan ketika *push button* berada pada posisi tidak ditekan, led indikator mati dan ketika *push button* berada pada posisi ditekan, led indikator menyala.

4.2.3 Analisis HC-05 Bluetooth Module

HC-05 *bluetooth module* dapat dikoneksikan baik dengan modul HC-05 lain ataupun perangkat yang memiliki fasilitas *bluetooth*. Jarak pengiriman ataupun penerimaan data dari HC-05 *bluetooth module* berkisar dari 1 – 10 meter. Koneksi *bluetooth* HC-05 akan terputus ketika jarak melebihi 10 meter. Kualitas koneksi juga dipengaruhi oleh media penghambat seperti dinding, pintu ataupun kaca jendela.

4.2.4 Analisis Perubahan Nilai Input Terhadap Nilai Output

Ketika terjadi perubahan nilai *input* baik dari sensor suhu ataupun *push button*, sistem pada *remote control* langsung melakukan eksekusi dengan membuat perhitungan logika *fuzzy* untuk menentukan nilai *output* berupa nilai *setting* suhu pada AC. Penentuan nilai *setting* suhu pada AC dengan menggunakan perhitungan logika *fuzzy* akan selalu dilakukan setiap sistem merespon adanya perubahan nilai *input* baik dari sensor suhu ataupun *push button*. Pada kondisi nilai *input* suhu luar sebesar 28 °C, nilai suhu udara didalam ruangan sebesar 25 °C dan jumlah penghuni adalah 1 orang, didapatkan hasil *output setting* AC sebesar 23 °C. Dengan perhitungan yang dilakukan secara manual, didapatkan variabel linguistik *input* suhu luar adalah sejuk dengan derajat keanggotaan $\frac{1}{7}$ dan normal dengan derajat keanggotaan $\frac{6}{7}$, *input* suhu dalam adalah normal dengan derajat keanggotaan $\frac{7}{8}$ dan hangat dengan derajat keanggotaan $\frac{1}{8}$, *input* jumlah penghuni adalah sepi dengan derajat keanggotaan $\frac{3}{5}$ dan sedang dengan derajat keanggotaan $\frac{2}{5}$. Pada proses inferensi, didapatkan hasil variabel linguistik

setting AC yaitu tinggi dengan derajat keanggotaan $\frac{3}{5}$ dan sedang dengan derajat keanggotaan $\frac{2}{5}$. Pada proses defuzzyfikasi didapatkan hasil *setting* ac seperti berikut.

$$y' = \frac{\frac{3}{5}(21) + \frac{2}{5}(25)}{\frac{3}{5} + \frac{2}{5}} = 13 + 10 = 23 \text{ } ^\circ\text{C}$$

didapatkan hasil sebesar 23 °C dari hasil perhitungan manual sehingga hasil perhitungan dari kontroler dan hasil hitung manual memiliki nilai yang sama.

4.3 Pembahasan

Berdasarkan hasil analisis data penelitian, didapatkan bahwa nilai pada sensor suhu dengan jenis *thermocouple type K* memiliki nilai pembacaan yang memiliki akurasi yang mendekati sebuah thermometer digital sehingga dapat digunakan sebagai alat pengukur suhu yang cukup baik. Pada penghitung (*counter*), ketika *push button 1* dan *push button 2* berada pada kondisi awal (tidak ditekan) maka memiliki nilai logika 1 karena *push button 1* dan *push button 2* dipasang pada posisi *pulldown switch* dan ketika *push button 1* dan *push button 2* berada pada kondisi ditekan maka nilai logika pada kedua *push button* akan berubah menjadi 0.

Dan pada sistem secara keseluruhan, Ketika terjadi perubahan nilai *input* baik dari sensor suhu ataupun *push button*, sistem pada *remote control* langsung melakukan eksekusi dengan membuat perhitungan logika *fuzzy* untuk menentukan nilai *output* berupa nilai *setting* suhu pada AC dan pengekseskuan akan selalu dilakukan ketika sistem merespon adanya perubahan pada nilai *input*.

4.4 Aplikasi Hasil Penelitian

Pengaplikasian dari produk penelitian yang telah dihasilkan adalah pada ruangan yang sering digunakan seperti ruang kelas atau ruang dosen untuk sebuah universitas ataupun ruang karyawan untuk sebuah perusahaan. Namun, masih ada beberapa kekurangan didalam alat yang dibuat, diantaranya :

1. Alat yang dibuat hanya bisa berkomunikasi secara satu arah, dimana *remote control* penerima hanya bisa menerima data dari *remote control* pengirim dan tidak bisa memberikan umpan balik (*feedback*) kepada *remote control* 2.
2. Suplai yang digunakan untuk *remote Air Conditioner* (AC) asli masih menggunakan suplai eksternal yaitu baterai AA yang membuat ruang didalam *box* menjadi sempit.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah alat dibuat dan diuji coba maka didapatkan kesimpulan :

Prototype Automatic Remote Control untuk *Air Conditioner* menggunakan Penerapan Metode Kontrol Logika *Fuzzy* berdasarkan Kondisi Suhu Udara Luar, Suhu Udara Didalam Ruangan dan Jumlah Penghuni Ruangan Berbasis Arduino Uno R3 berhasil dibuat dan hasil pengujian menunjukkan bahwa *Prototype Automatic Remote Control* untuk *Air Conditioner* menggunakan Penerapan Metode Kontrol Logika *Fuzzy* berdasarkan Kondisi Suhu Udara Luar, Suhu Udara Didalam Ruangan dan Jumlah Penghuni Ruangan Berbasis Arduino Uno R3 dapat digunakan untuk merubah nilai *setting* pada AC menggunakan perhitungan logika *fuzzy* berdasarkan perubahan kondisi nilai suhu ataupun perubahan jumlah penghuni didalam ruangan secara otomatis.

5.2 Saran

Berdasarkan kesimpulan diatas, maka saran yang dapat diberikan oleh penulis untuk mengembangkan *prototype* yang telah dibuat adalah :

1. Penggunaan *bluetooth* dapat diganti menggunakan modul Wi-Fi dan dimonitoring melalui LAN atau internet
2. Membuat komunikasi antar *remote control* menjadi 2 arah atau membuat hanya satu buah *remote control* yang bisa mengendalikan *Air Conditioner* (AC) secara langsung tanpa menggunakan perantara *remote AC* asli

DAFTAR PUSTAKA

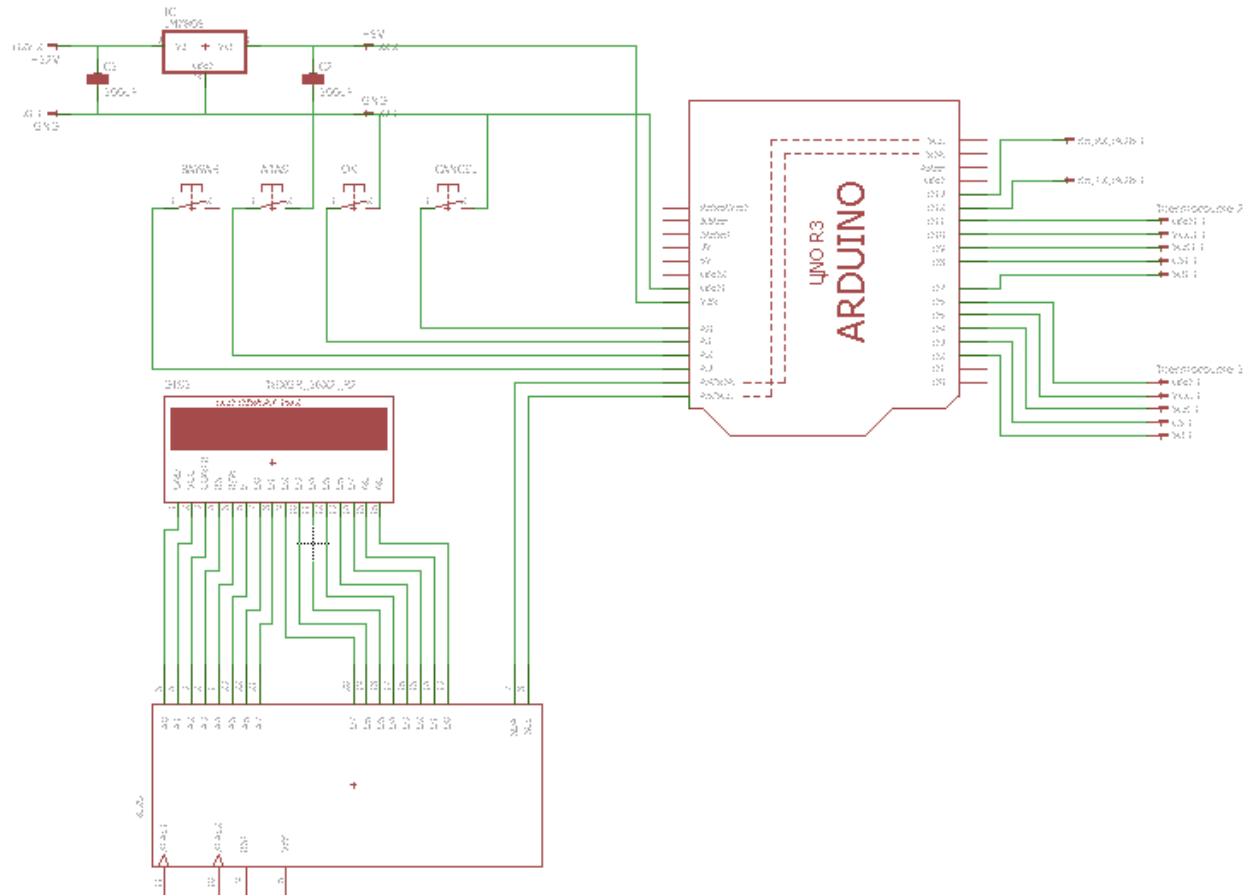
- Arief Gozhali, Fanani. (2015). *Penggunaan Logika Fuzzy Untuk Pengembangan Instrumen Penilaian Kinerja Praktik Sistem Kontrol Terprogram Siswa Sekolah Menengah Kejuruan*. Universitas Negeri Yogyakarta
- Fadholi, Akhmad. (2013). *Study Pengaruh Suhu dan Tekanan Udara Terhadap Operasi Penerbangan di Bandara H.A.S Hananjoeddin Buluh Tumbang Belitung Periode 1980 - 2010*. *Jurnal Penelitian Fisika Dan Aplikasinya*, Vol.3 No.1.
- Iswandy, Eka. (2015). *Sistem Penunjang Keputusan Untuk Menentukan Penerimaan Dana Santunan Sosial Anak Nagari Dan Penyalurannya Bagi Mahasiswa Dan Pelajar Kurang Mampu Di Kenagarian Barung-Barung Balantai Timur*. *Jurnal TEKNOIF*, Vol.3, No.2.
- Mayangsari, Maudi. (2016).[Skripsi]. *Analisis Fuzzy Logic Pada Pengaruh Suhu Udara Luar dan Jumlah Manusia Terhadap Suhu Setting Air Conditioner (AC) di Gedung L1 Jurusan Teknik Elektro Fakultas Teknik Universitas Negeri Jakarta*. Universitas Negeri Jakarta.
- Naba, Agus. 2009. *Belajar Cepat Fuzzy Logic Menggunakan MATLAB*. Yogyakarta : Penerbit Andi
- Nasution, Helfi. (2012). *Implementasi Logika Fuzzy pada Sistem Kecerdasan Buatan*. *Jurnal ELKHA*, Vol.4, No.2.
- Rachmad, Anggi. (2016).[Skripsi]. *Rancang Bangun Robot Pemanjat Tiang dan Pemasang Propeller Menggunakan Gripper Elektropneumatik Sebagai Hybrid Robot Pada Kontes Robot ABU Indonesia 2016*. Universitas Negeri Jakarta
- Saputra, Rusli. (2015). *Desain Sistem Informasi Order Photo Pada Creative Studio Photo Dengan Menggunakan Bahasa Pemrograman Visual Basic.Net 2010*. *Jurnal Momentum*, Vol.17, No.2.
- [Kemendikbud] Kementerian Pendidikan Dan Kebudayaan Republik Indonesia. (2013). *Sistem dan Instalasi Tata Udara*.

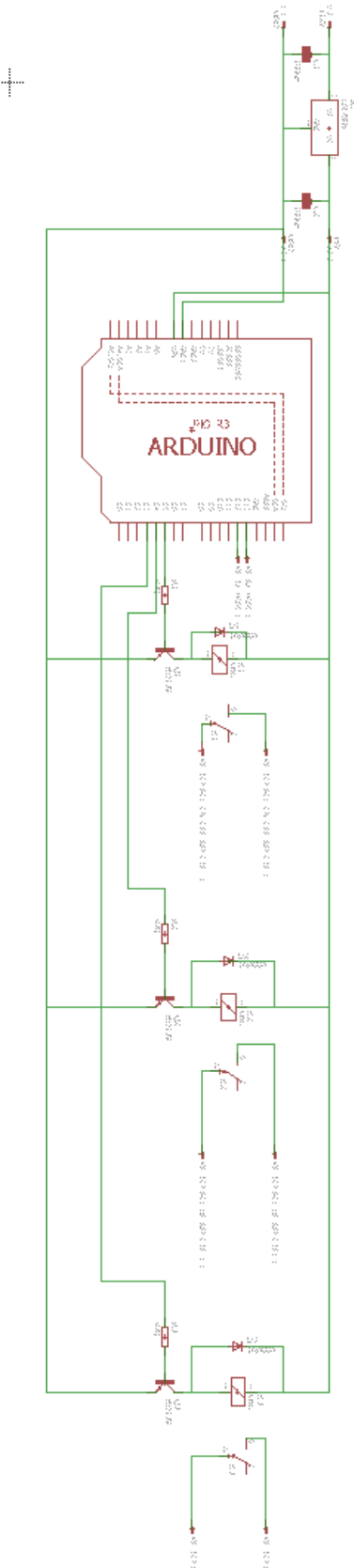
LAMPIRAN-LAMPIRAN

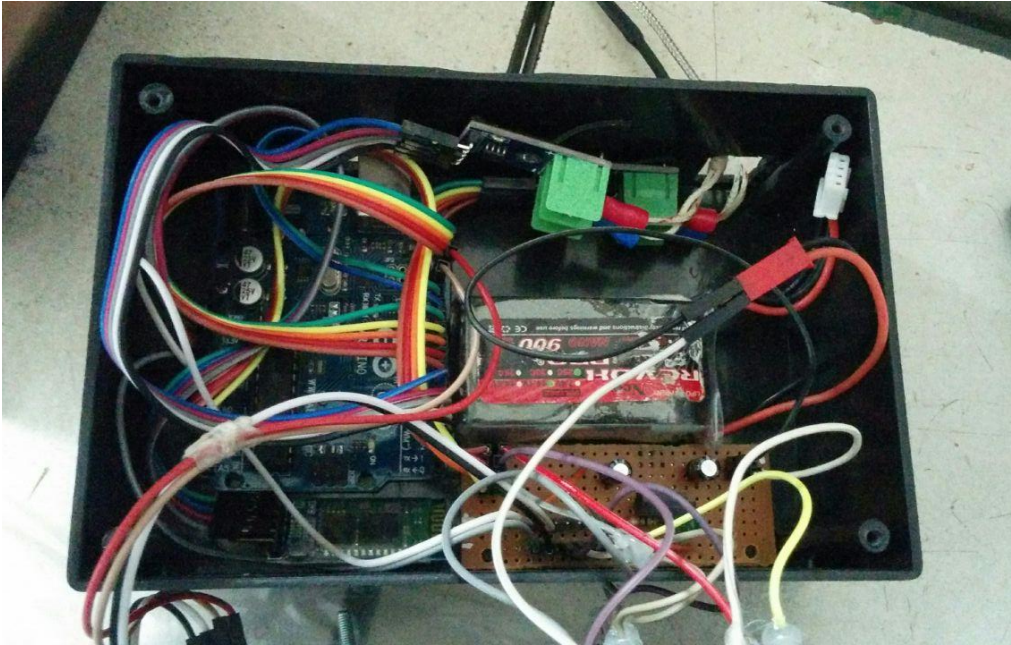
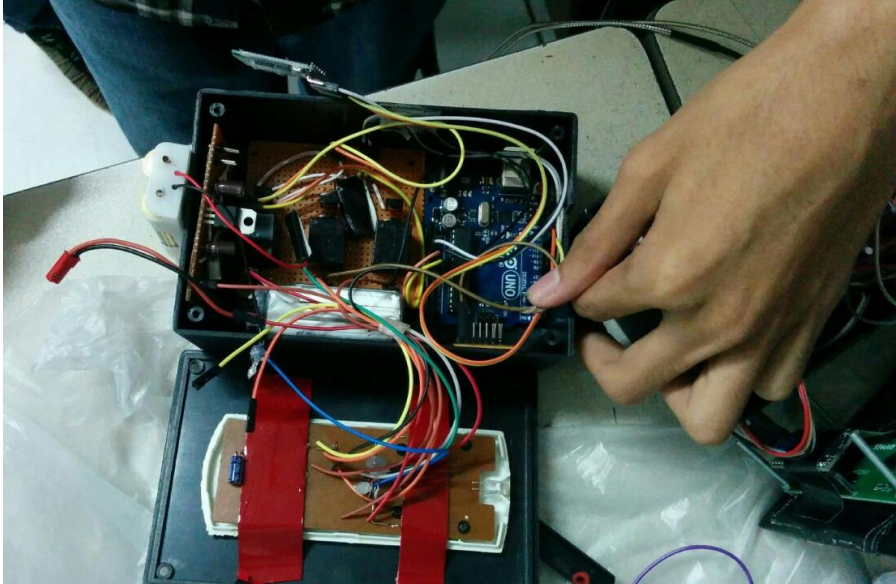
LAMPIRAN 1. Dokumentasi Produk Yang Dihasilkan



LAMPIRAN 2. Skematik Rangkaian







LAMPIRAN 3 Data-Data Pengukuran

1. Pengukuran Tegangan Pada *Push Button (Pull Down Resistor)*

No	Kondisi <i>Push Button</i>	Tegangan	Indikator
1	Ditekan	4,8 volt	<i>ON</i>
2	Tidak Ditekan	4,8 volt	<i>OFF</i>

2. Pengukuran Tegangan Pada *Push Button (Pull Up Resistor)*

No	Kondisi <i>Push Button</i>	Tegangan	Indikator
1	Ditekan	4,8 volt	<i>OFF</i>
2	Tidak Ditekan	4,8 volt	<i>ON</i>

3. Pengukuran Tegangan Pada Basis Transistor 1

No	Logika Pada Basis	Tegangan
1	<i>LOW</i>	0 volt
2	<i>HIGH</i>	4,5 volt

4. Pengukuran Tegangan Pada Basis Transistor 2

No	Logika Pada Basis	Tegangan
1	<i>LOW</i>	0 volt
2	<i>HIGH</i>	4,5 volt

5. Pengukuran Tegangan Pada Basis Transistor 3

No	Logika Pada Basis	Tegangan
1	<i>LOW</i>	0 volt
2	<i>HIGH</i>	4,5 volt

LAMPIRAN 4 Data-Data Perhitungan

1. Perhitungan Nilai Kesalahan (*Error*) Pada Pembacaan Sensor

Thermocouple

Perhitungan nilai kesalahan (*error*) pada pembacaan sensor *thermocouple* ketika dikalibrasi dengan *thermometer digital* adalah :

$$Error = \frac{NTD - NTC}{NTC} \times 100\%$$

Keterangan :


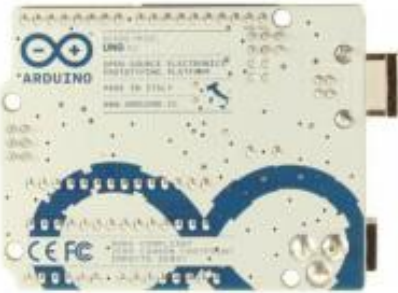
NTD : Nilai suhu yang terbaca pada *thermometer digital*


NTC : Nilai suhu yang terbaca pada sensor *thermocouple*

LAMPIRAN 5 Data Pendukung Lain Yang Berkaitan


1. Datasheet Arduino Uno R3

Arduino Uno








Arduino Uno R2 Front



Arduino Uno SMD



Arduino Uno Front



Arduino Uno Back

Overview

The Arduino Uno is a microcontroller board based on the ATmega328 ([datasheet](#)). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

[Revision 2](#) of the Uno board has a resistor pulling the BU2 HWB line to ground, making it easier to put into [DFU mode](#).

[Revision 3](#) of the board has the following new features:

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

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

Summary

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

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

Schematic & Reference Design

EAGLE files: [arduino-uno-Rev3-reference-design.zip](#) (NOTE: works with Eagle 6.0 and newer)

Schematic: [arduino-uno-Rev3-schematic.pdf](#)

Note: The Arduino reference design can use an Atmega8, 168, or 328. Current models use an ATmega328, but an Atmega8 is shown in the schematic for reference. The pin configuration is identical on all three processors.

Power

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

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

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

The power pins are as follows:

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

Memory

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

Input and Output

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

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

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

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

- **TWI: A4 or SDA pin and A5 or SCL pin.** Support TWI communication using the [Wire library](#).

There are a couple of other pins on the board:

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

See also the [mapping between Arduino pins and ATmega328 ports](#). The mapping for the ATmega8, 168, and 328 is identical.

Communication

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

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

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

Programming

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

The ATmega328 on the Arduino Uno comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

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

The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available. The ATmega16U2/8U2 is loaded with a DFU bootloader, which can be activated by:

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

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

Automatic (Software) Reset

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

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

USB Overcurrent Protection

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

Physical Characteristics

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

2. Datasheet MAX6675 Module

Manual MAX6675 K-Type Thermocouple Temperature Sensor

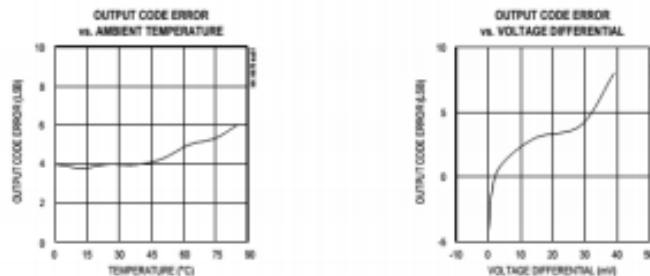
Gambaran Umum

MAX6675 dibentuk dari kompensasi cold-junction yang outputnya didigitalisasi dari sinyal termokopel tipe-K. data output memiliki resolusi 12-bit dan mendukung komunikasi SPI mikrocontroller secara umum. Data dapat dibaca dengan mengkonversi hasil pembacaan 12-bit data.

Fitur

- Konversi digital langsung dari output termokopel tipe-K
- Kompensasi cold-junction
- Komunikasi kompatibel dengan protocol SPI
- Open thermocouple detection

Karakteristik



(sumber datasheet MAX6675)

Cold-Junction Compensation

Fungsi dari termokopel adalah untuk mengetahui perbedaan temperature di bagian ujung dari dua bagian metal yang berbeda dan disatukan. Termokopel tipe *hot junction* dapat mengukur mulai dari 0°C sampai +1023,75°C. MAX6675 memiliki bagian ujung *cold end* yang hanya dapat mengukur -20°C sampai +85°C. Pada saat bagian *cold end* MAX6675 mengalami fluktuasi suhu maka MAX6675 akan tetap dapat mengukur secara akurat perbedaan temperature pada bagian yang lain. MAX6675 dapat melakukan koreksi atas perubahan pada temperature ambient dengan kompensasi *cold-junction*. Device mengkonversi temperature ambient yang terjadi ke bentuk tegangan menggunakan sensor temperature diode. Untuk dapat melakukan pengukuran actual, MAX6675 mengukur tegangan dari output termokopel dan tegangan dari sensing diode.

Performance optimal MAX6675 dapat tercapai pada waktu termokopel bagian *cold-junction* dan MAX6675 memiliki temperature yang sama. Hal ini untuk menghindari penempatan komponen lain yang menghasilkan panas didekat MAX6675.

Manual MAX6675 K-Type Thermocouple Temperature Sensor

Konversi Temperatur

Untuk mendapatkan nilai suhu pada termokopel tipe-K, perubahan tegangan sebesar $41\mu\text{V}/^\circ\text{C}$ dengan menggunakan pendekatan karakteristik dapat menggunakan rumus dibawah ini:

$$V_{OUT} = (41\mu\text{V} / ^\circ\text{C}) S (T_k - T_{AMB})$$

Dengan:

V_{OUT} = Voutput termokopel (μV)

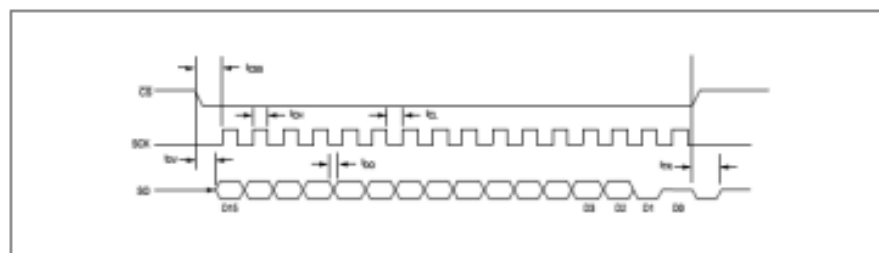
T_k adalah temperatur remote junction ($^\circ\text{C}$)

T_{AMB} adalah temperature ambient ($^\circ\text{C}$)

Timing Clock



Protokol Interfacing Serial



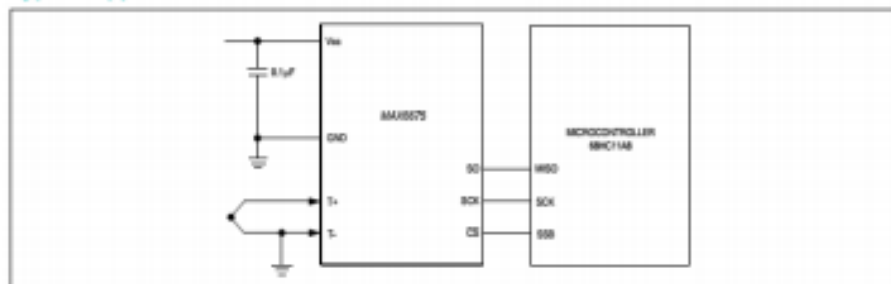
Timing Interfacing Serial

Manual MAX6675 K-Type Thermocouple Temperature Sensor

BIT	DUMMY SIGN BIT	12-BIT TEMPERATURE READING											THERMOCOUPLE INPUT	DEVICE ID	STATE	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	MSB											LSB		0	Three- state

SO Output

Pemasangan Modul



Hubungan pin antara mikrokontroler dengan modul MAX6675

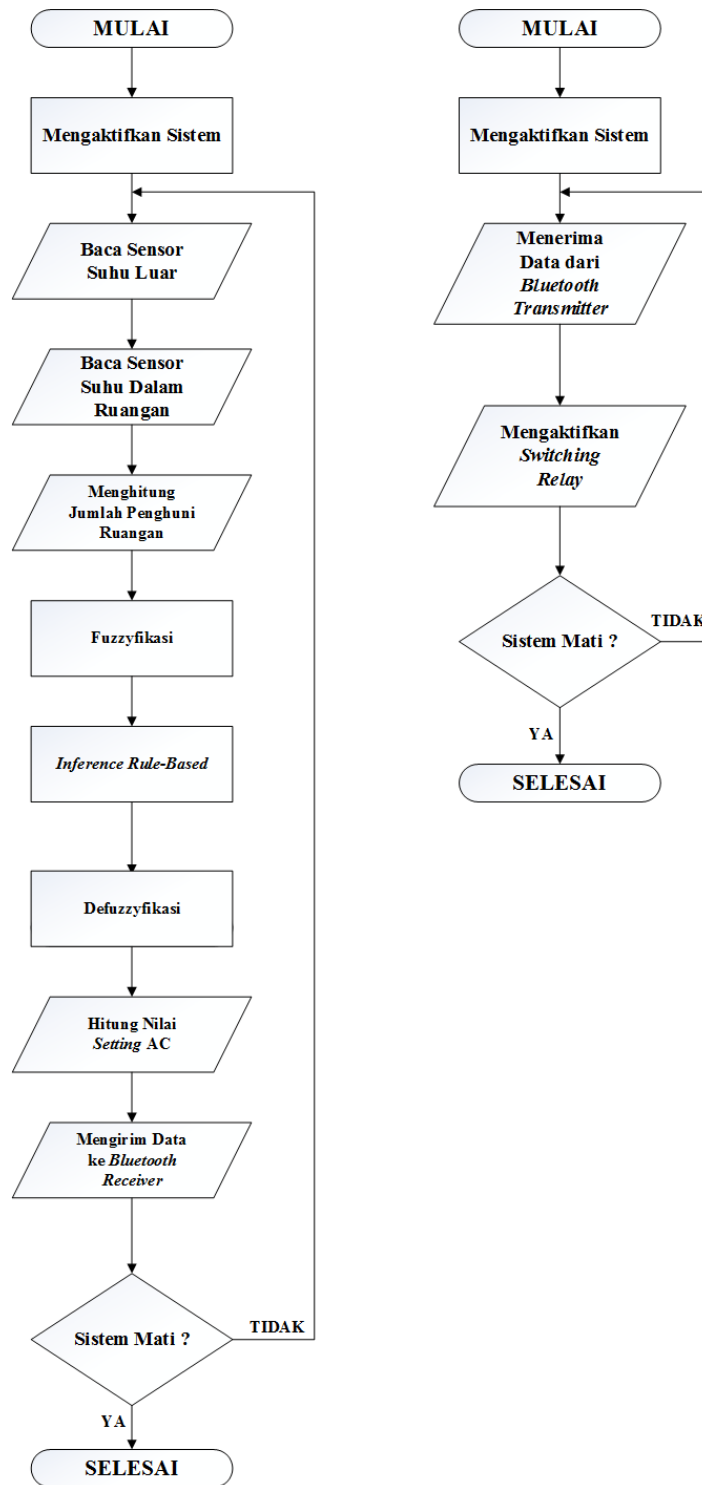
Modul MAX6675	Mikrokontroler ATmega32
VCC	+5V
SO	MISO (PB6)
CS	SS (PB4)
CSK	SCK (PB7)
GND	GND

Program Testing

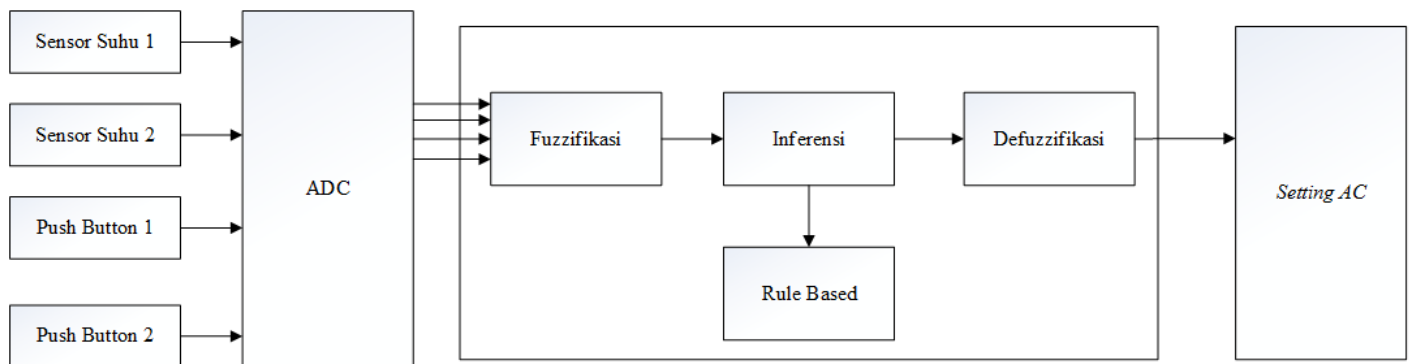
Sistem yang digunakan:

- Sistem minimum mikrokontroler ATmega32
- Modul LCD 16x2
- Modul Sensor Temperatur Termokopel tipe-K MAX6675

3. Flowchart Alat



4. Blok Diagram Sistem



5. Datasheet HC-05 Bluetooth Module



HC-05

-Bluetooth to Serial Port Module

Overview



HC-05 module is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup.

Serial port Bluetooth module is fully qualified Bluetooth V2.0+EDR (Enhanced Data Rate) 3Mbps Modulation with complete 2.4GHz radio transceiver and baseband. It uses CSR Bluecore 04-External single chip Bluetooth system with CMOS technology and with AFH(Adaptive Frequency Hopping Feature). It has the footprint as small as 12.7mmx27mm. Hope it will simplify your overall design/development cycle.

Specifications

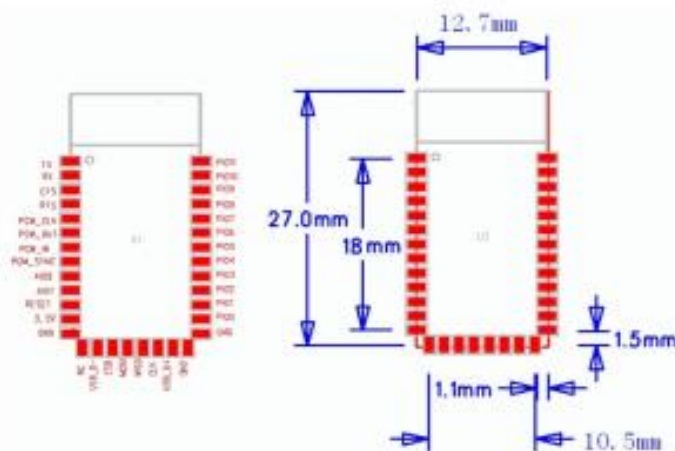
Hardware features

- Typical +80dBm sensitivity
- Up to +4dBm RF transmit power
- Low Power 1.8V Operation ,1.8 to 3.6V I/O
- PIO control
- UART interface with programmable baud rate
- With integrated antenna
- With edge connector

Software features

- Default Baud rate: 38400, Data bits:8, Stop bit:1,Parity:No parity, Data control: has. Supported baud rate: 9600,19200,38400,57600,115200,230400,460800.
- Given a rising pulse in PIO0, device will be disconnected.
- Status instruction port PIO1: low-disconnected, high-connected;
- PIO10 and PIO11 can be connected to red and blue led separately. When master and slave are paired, red and blue led blinks 1time/2s in interval, while disconnected only blue led blinks 2times/s.
- Auto-connect to the last device on power as default.
- Permit pairing device to connect as default.
- Auto-pairing PINCODE:"0000" as default
- Auto-reconnect in 30 min when disconnected as a result of beyond the range of connection.

Hardware



PIN Name	PIN #	Pad type	Description	Note
GND	13	VSS	Ground pot	
	21			
	22			
3.3 VCC	12	3.3V	Integrated 3.3V (+) supply with On-chip linear regulator output within 3.15-3.3V	
AIO0	9	Bi-Directional	Programmable input/output line	
AIO1	10	Bi-Directional	Programmable input/output line	
PIO0	23	Bi-Directional RX EN	Programmable input/output line, control output for LNA(if fitted)	
PIO1	24	Bi-Directional TX EN	Programmable input/output line, control output for PA(if fitted)	
PIO2	25	Bi-Directional	Programmable input/output line	
PIO3	26	Bi-Directional	Programmable input/output line	
PIO4	27	Bi-Directional	Programmable input/output line	
PIO5	28	Bi-Directional	Programmable input/output line	
PIO6	29	Bi-Directional	Programmable input/output line	
PIO7	30	Bi-Directional	Programmable input/output line	
PIO8	31	Bi-Directional	Programmable input/output line	
PIO9	32	Bi-Directional	Programmable input/output line	
PIO10	33	Bi-Directional	Programmable input/output line	
PIO11	34	Bi-Directional	Programmable input/output line	

RESETB	11	CMOS input with weak internal pull-up	Reset if low, input debounced so must be low for >5MS to cause a reset	
UART_RTS	4	CMOS output, tri-stable with weak internal pull-up	UART request to send, active low	
UART_CTS	3	CMOS input with weak internal pull-down	UART clear to send, active low	
UART_RX	2	CMOS input with weak internal pull-down	UART Data input	
UART_TX	1	CMOS output, Tri-stable with weak internal pull-up	UART Data output	
SPI_MOSI	17	CMOS input with weak internal pull-down	Serial peripheral interface data input	
SPI_CSB	16	CMOS input with weak internal pull-up	Chip select for serial peripheral interface, active low	
SPI_CLK	19	CMOS input with weak internal pull-down	Serial peripheral interface clock	
SPI_MISO	18	CMOS input with weak internal pull-down	Serial peripheral interface data Output	
USB_	15	Bi-Directional		

6. Listing Program

A. Master Program

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

LiquidCrystal_I2C lcd(0x3F, 16, 2);

// ThermoCouple 1
int thermo1_so_pin = 2;
int thermo1_cs_pin = 3;
int thermo1_sck_pin = 4;
int thermo1_vcc_pin = 5;
int thermo1_gnd_pin = 6;

// ThermoCouple 2
int thermo2_so_pin = 8;
int thermo2_cs_pin = 9;
int thermo2_sck_pin = 10;
int thermo2_vcc_pin = 11;
int thermo2_gnd_pin = 12;

Scan_Suhu Scan_1 (thermo1_sck_pin, thermo1_cs_pin, thermo1_so_pin);
Scan_Suhu Scan_2 (thermo2_sck_pin, thermo2_cs_pin, thermo2_so_pin);

#define bawah A0
#define atas A1
#define batal A2
#define oke A3

//Loading Awal
int loading;
int stop_loading = 100;
int masuk_menu;

//Off Mode
int off = 999;
```

```
//Main Menu
int main_menu;
int pilih_menu;

//Menu 1 (Pilih Mode)
int menu_1_geser;
int menu_1;
int pilih_mode;

//Pilih Mode (Manual)
int mode_manual;

//Pilih Mode (Auto)
int mode_auto;

//Menu 2 (Scan Suhu)
int menu_2_geser;
int menu_2;
int pilih_sensor;
int TC_1;
int TC_2;

void setup() {
  // put your setup code here, to run once:

  //Setup Pin
  digitalWrite(bawah, HIGH);
  digitalWrite(atas, HIGH);
  digitalWrite(batal, HIGH);
  digitalWrite(oke, HIGH);

  pinMode(thermo1_vcc_pin, OUTPUT);
  pinMode(thermo1_gnd_pin, OUTPUT);
  digitalWrite(thermo1_vcc_pin, HIGH);
  digitalWrite(thermo1_gnd_pin, LOW);

  pinMode(thermo2_vcc_pin, OUTPUT);
  pinMode(thermo2_gnd_pin, OUTPUT);
```

```

digitalWrite(thermo2_vcc_pin, HIGH);
digitalWrite(thermo2_gnd_pin, LOW);

//Setup LCD
lcd.init();
lcd.backlight();

//Tampilan Awal
lcd.setCursor(0, 0);
lcd.print("  Logic Os  ");
delay(1000);
lcd.setCursor(0, 1);
lcd.print("  Present... ");
delay(700);
hapus();
}

/*****/
void hapus()
{
  lcd.clear();
}

/*****/

void loop() {
  // put your main code here, to run repeatedly:
  if (masuk_menu == 0)
  {
    for (loading = 0; loading <= 16; loading++)
    {
      lcd.setCursor(0, 0);
      lcd.print("  On Boot... ");
      delay(20);
    }

    if (loading = 16)
    {
      hapus();
    }
  }
}

```

```
    delay(100);
    masuk_menu = 1;
    loading = stop_loading;
  }
}

//Masuk Main Menu
if (masuk_menu == 1)
{
  Main_Menu();
}

//Menu_1
if (menu_1 == 1)
{
  Menu_1();
}

//Menu_1 (Normal_Mode)
if (mode_manual == 1)
{
  Mode_Manual();
}

//Menu_1 (Fuzzy Mode)
if (mode_auto == 1)
{
  Mode_Fuzzy();
}

//Menu_2
if (menu_2 == 1)
{
  Menu_2();
}

//Menu_2 TC_1
if (TC_1 == 1)
{
```

```
        TC1());
    }

    //Menu_2 TC_2
    if (TC_2 == 1)
    {
        TC2();
    }
    if (digitalRead(batal) == LOW)
    {
        kembali_ke_menu_awal();
    }
}

void Main_Menu()
{
    switch (main_menu)
    {
        case 0:
            lcd.setCursor(0, 0);
            lcd.print("1. Pilih Mode ");
            pilih_menu = 1;
            break;
        case 1:
            lcd.setCursor(0, 0);
            lcd.print("2. Thermocouple ");
            pilih_menu = 2;
            break;
    }

    //Go To Menu 1
    if (digitalRead(oke) == 0 && pilih_menu == 1)
    {
        menu_1 = 1;
        masuk_menu = off;
        delay(100);
    }
}
```

```
if (digitalRead(oke) == 0 && pilih_menu == 2)
{
    menu_2 = 1;
    masuk_menu = off;
    delay(100);
}
/*****/

if (digitalRead(atas) == 0)
{
    main_menu -= 1;
    delay(300);
}

if (digitalRead(bawah) == 0)
{
    main_menu += 1;
    delay(300);
}

if (main_menu < 0)
{
    main_menu = 1;
}

if (main_menu > 1)
{
    main_menu = 0;
}
}

void kembali_ke_menu_awal()
{
    masuk_menu = 1;
    pilih_menu = 0;
    menu_1 = 0;
    pilih_mode = 0;
    mode_manual = 0;
    mode_auto = 0;
}
```



```

menu_2 = 0;
TC_1 = 0;
TC_2 = 0;
pilih_sensor = 0;
main_menu = 0;
menu_1_geser = 0;
menu_2_geser = 0;
}
void Menu_1()
{
  switch (menu_1_geser)
  {
    case 0:
      lcd.setCursor(0, 0);
      lcd.print("1. Mode Manual ");
      pilih_mode = 1;
      break;
    case 1:
      lcd.setCursor(0, 0);
      lcd.print("2. Mode Auto ");
      pilih_mode = 2;
      break;
  }

  //Go To Mode Manual
  if (pilih_mode == 1 && digitalRead(oke) == 0)
  {
    menu_1 = off;
    mode_manual = 1;
    lcd.clear();
    delay(100);
  }

  //Go To Mode Auto
  if (pilih_mode == 2 && digitalRead(oke) == 0)
  {
    menu_1 = off;
    mode_auto = 1;
    lcd.clear();
  }
}

```

```

    delay(100);
}

/*****/
if (digitalRead(atas) == 0)
{
    menu_1_geser -= 1;
    delay(500);
}

if (digitalRead(bawah) == 0)
{
    menu_1_geser += 1;
    delay(500);
}

if (menu_1_geser < 0)
{
    menu_1_geser = 1;
}

if (menu_1_geser > 1)
{
    menu_1_geser = 0;
}
}

int geser_nilai_mode_manual;

void Mode_Manual()
{
    lcd.setCursor(0,0);
    lcd.print("Setting AC : ");
    lcd.setCursor(13,0);
    lcd.print(geser_nilai_mode_manual);

/*****/
if (digitalRead(atas) == 0)
{

```

```
geser_nilai_mode_manual -= 1;
delay(500);
}

if (digitalRead(bawah) == 0)
{
geser_nilai_mode_manual += 1;
delay(500);
}

if (geser_nilai_mode_manual < 16)
{
geser_nilai_mode_manual = 16;
}

if (geser_nilai_mode_manual > 30)
{
geser_nilai_mode_manual = 30;
}
}

float baca_sensor_suhu_dalam;
float baca_sensor_suhu_luar;
float ukur_suhu_dalam;
float ukur_suhu_luar;

float jumlah_penghuni;
float hitung_jumlah_penghuni;

float hasil;

float suhu_luar1;
float suhu_luar2;
float suhu_luar3;
float suhu_luar4;
float suhu_luar5;

float suhu_dalam1;
float suhu_dalam2;
```

```
float suhu_dalam3;
float suhu_dalam4;
float suhu_dalam5;

float jumlah_penghunil ;
float jumlah_penghuni2;
float jumlah_penghuni3;
float jumlah_penghuni4;
float jumlah_penghuni5;

//suhu luar
int sejuk;
int normal;
int hangat;

//suhu dalam
int sejuk_dalam;
int normal_dalam;
int hangat_dalam;

//jumlah orang
int sepi;
int sedang_ ;
int cukup_ramai;
int padat;
int sangat_padat;

float rules1; float rules10; float rules19; float rules28; float rules37;
float rules2; float rules11; float rules20; float rules29; float rules38;
float rules3; float rules12; float rules21; float rules30; float rules39;
float rules4; float rules13; float rules22; float rules31; float rules40;
float rules5; float rules14; float rules23; float rules32; float rules41;
float rules6; float rules15; float rules24; float rules33; float rules42;
float rules7; float rules16; float rules25; float rules34; float rules43;
float rules8; float rules17; float rules26; float rules35; float rules44;
float rules9; float rules18; float rules27; float rules36; float rules45;

float set1; float set10; float set19; float set28; float set37;
float set2; float set11; float set20; float set29; float set38;
```

float set3; float set12; float set21; float set30; float set39;
float set4; float set13; float set22; float set31; float set40;
float set5; float set14; float set23; float set32; float set41;
float set6; float set15; float set24; float set33; float set42;
float set7; float set16; float set25; float set34; float set43;
float set8; float set17; float set26; float set35; float set44;
float set9; float set18; float set27; float set36; float set45;

float rendah1; float rendah5; float rendah9; float rendah13; float
rendah17;
float rendah2; float rendah6; float rendah10; float rendah14; float
rendah18;
float rendah3; float rendah7; float rendah11; float rendah15; float
rendah19;
float rendah4; float rendah8; float rendah12; float rendah16; float
rendah20;

float sedang1; float sedang5; float sedang9; float sedang13; float
sedang17;
float sedang2; float sedang6; float sedang10; float sedang14; float
sedang18;
float sedang3; float sedang7; float sedang11; float sedang15; float
sedang19;
float sedang4; float sedang8; float sedang12; float sedang16; float
sedang20;

float tinggi1; float tinggi5; float tinggi9; float tinggi13; float tinggi17;
float tinggi2; float tinggi6; float tinggi10; float tinggi14; float tinggi18;
float tinggi3; float tinggi7; float tinggi11; float tinggi15; float tinggi19;
float tinggi4; float tinggi8; float tinggi12; float tinggi16; float tinggi20;

float maximum_tinggi1;
float maximum_tinggi2;
float maximum_tinggi3;
float maximum_tinggi4;
float maximum_tinggi5;
float maximum_tinggi6;
float maximum_tinggi7;
float maximum_tinggi8;

```
float maximum_tinggi9;  
float maximum_tinggi10;  
float maximum_tinggi11;
```

```
float maximum_sedang1;  
float maximum_sedang2;  
float maximum_sedang3;  
float maximum_sedang4;  
float maximum_sedang5;  
float maximum_sedang6;  
float maximum_sedang7;  
float maximum_sedang8;  
float maximum_sedang9;  
float maximum_sedang10;  
float maximum_sedang11;  
float maximum_sedang12;  
float maximum_sedang13;  
float maximum_sedang14;  
float maximum_sedang15;
```

```
float maximum_rendah1;  
float maximum_rendah2;  
float maximum_rendah3;  
float maximum_rendah4;  
float maximum_rendah5;  
float maximum_rendah6;  
float maximum_rendah7;  
float maximum_rendah8;  
float maximum_rendah9;  
float maximum_rendah10;  
float maximum_rendah11;  
float maximum_rendah12;  
float maximum_rendah13;  
float maximum_rendah14;  
float maximum_rendah15;  
float maximum_rendah16;
```

```
float output_1 = 16;  
float output_2 = 23;
```

```
float output_3 = 30;

unsigned long waktu1;
unsigned long waktu2;
unsigned long delay1;
unsigned long delay2;
int interval_data = 30000;
int interval = 500;

int set_to_default = 25;

void Mode_Fuzzy()
{

    counter();
    waktu1 = millis();
    delay1 = millis();

    if(delay1 - delay2 >= interval_data)
    {
        delay2 = delay1;
        Serial.write(int(hasil));

        if (hasil == 0)
        {
            hasil = set_to_default;
        }
    }

    if(waktu1 - waktu2 >= interval)
    {
        waktu2 = waktu1;

        baca_sensor_suhu_luar = Scan_1.Celsius();
        baca_sensor_suhu_dalam = Scan_2.Celsius();
        ukur_suhu_luar = baca_sensor_suhu_luar;
        ukur_suhu_dalam = baca_sensor_suhu_dalam;

        if (ukur_suhu_luar < 22)
```

```
{
    ukur_suhu_luar = 22;
}

if (ukur_suhu_luar > 36)
{
    ukur_suhu_luar = 36;
}

if (ukur_suhu_dalam < 22)
{
    ukur_suhu_dalam = 22;
}

if (ukur_suhu_dalam > 36)
{
    ukur_suhu_dalam = 36;
}

fuzzyfikasi();
rules();
defuzzyfikasi();
hasil();

lcd.setCursor(0, 0);
lcd.print("SD:");
lcd.setCursor(4, 0);
lcd.print(ukur_suhu_dalam, 1);
lcd.setCursor(8, 0);
lcd.print("SL:");
lcd.setCursor(11, 0);
lcd.print(ukur_suhu_luar, 1);
lcd.setCursor(0, 1);
lcd.print("JP:");
lcd.setCursor(4, 1);
lcd.print(hitung_jumlah_penghuni, 0);
lcd.setCursor(8, 1);
lcd.print("AC:");
lcd.setCursor(12, 1);
```



```
lcd.print(hasil, 0);
}
}

void counter()
{
  if (digitalRead(bawah) == 0)
  {
    hitung_jumlah_penghuni = hitung_jumlah_penghuni + 1;
    delay(500);
  }

  if (digitalRead(atas) == 0)
  {
    hitung_jumlah_penghuni = hitung_jumlah_penghuni - 1;
    delay(500);
  }

  if (hitung_jumlah_penghuni < 0)
  {
    hitung_jumlah_penghuni = 0;
  }

  if (hitung_jumlah_penghuni > 10)
  {
    hitung_jumlah_penghuni = 10;
  }

  if (hitung_jumlah_penghuni < 10)
  {
    lcd.setCursor(5, 1);
    lcd.print(" ");
  }
}
```

```
void fuzzyfikasi()
{
  /*******Suhu Luar*****/
  if (ukur_suhu_luar <= 22)
  {
    suhu_luar1 = 1;
    sejuk = 1;
  }

  else if (ukur_suhu_luar > 22 && ukur_suhu_luar < 29)
  {
    suhu_luar1 = (29 - ukur_suhu_luar) / (29 - 22);
    sejuk = 1;
  }

  else if (ukur_suhu_luar >= 29)
  {
    suhu_luar1 = 0;
    sejuk = 1;
  }

  //kondisi normal
  if (ukur_suhu_luar <= 22)
  {
    suhu_luar2 = 0;
    normal = 1;
  }

  else if (ukur_suhu_luar > 22 && ukur_suhu_luar < 29)
  {
    suhu_luar2 = (ukur_suhu_luar - 22) / (29 - 22);
    normal = 1;
  }

  else if (ukur_suhu_luar == 29)
  {
    suhu_luar2 = 1;
    normal = 1;
  }
}
```

```
else if (ukur_suhu_luar > 29 && ukur_suhu_luar < 36)
{
    suhu_luar2 = (36 - ukur_suhu_luar) / (36 - 29);
    normal = 1;
}

else if (ukur_suhu_luar >= 36)
{
    suhu_luar2 = 0;
    normal = 1;
}

//kondisi hangat
if (ukur_suhu_luar <= 29)
{
    suhu_luar3 = 0;
    hangat = 1;
}

else if (ukur_suhu_luar > 29 && ukur_suhu_luar < 36)
{
    suhu_luar3 = (ukur_suhu_luar - 29) / (36 - 29);
    hangat = 1;
}

else if (ukur_suhu_luar >= 36)
{
    suhu_luar3 = 1;
    hangat = 1;
}

/*****Suhu Dalam*****/
//kondisi sejuk

if (ukur_suhu_dalam <= 20)
{
```

```
suhu_dalam1 = 1;
sejuk_dalam = 1;
}

else if (ukur_suhu_dalam > 20 && ukur_suhu_dalam < 26)
{
    suhu_dalam1 = (26 - ukur_suhu_dalam) / (26 - 20);
    sejuk_dalam = 1;
}

else if (ukur_suhu_dalam >= 26)
{
    suhu_dalam1 = 0;
    sejuk_dalam = 1;
}

//kondisi normal
if (ukur_suhu_dalam <= 20)
{
    suhu_dalam2 = 0;
    normal_dalam = 1;
}

else if (ukur_suhu_dalam > 20 && ukur_suhu_dalam < 26)
{
    suhu_dalam2 = (ukur_suhu_dalam - 20) / (26 - 20);
    normal_dalam = 1;
}

else if (ukur_suhu_dalam == 26)
{
    suhu_dalam2 = 1;
    normal_dalam = 1;
}

else if (ukur_suhu_dalam > 26 && ukur_suhu_dalam < 32)
{
    suhu_dalam2 = (32 - ukur_suhu_dalam) / (32 - 26);
    normal_dalam = 1;
}
```

```
else if (ukur_suhu_dalam >= 32)
{
    suhu_dalam2 = 0;
    normal_dalam = 1;
}

//kondisi hangat
if (ukur_suhu_dalam <= 26)
{
    suhu_dalam3 = 0;
    hangat_dalam = 1;
}

else if (ukur_suhu_dalam > 26 && ukur_suhu_dalam < 32)
{
    suhu_dalam3 = (ukur_suhu_dalam - 26) / (32 - 26);
    hangat_dalam = 1;
}

else if (ukur_suhu_dalam >= 32)
{
    suhu_dalam3 = 1;
    hangat_dalam = 1;
}

/*****Jumlah Penghuni*****/
//kondisi sepi
if (hitung_jumlah_penghuni <= 0)
{
    jumlah_penghunil = 1;
    sepi = 1;
}

else if (hitung_jumlah_penghuni > 0 && hitung_jumlah_penghuni < 2.5)
{
    jumlah_penghunil = (2.5 - hitung_jumlah_penghuni) / (2.5 - 0);
    sepi = 1;
}
```

```
else if (hitung_jumlah_penghuni >= 2.5)
{
    jumlah_penghuni1 = 0;
    sepi = 1;
}

//kondisi sedang
if (hitung_jumlah_penghuni <= 0)
{
    jumlah_penghuni2 = 0;
    sedang_ = 1;
}

else if (hitung_jumlah_penghuni > 0 && hitung_jumlah_penghuni < 2.5)
{
    jumlah_penghuni2 = (hitung_jumlah_penghuni - 0) / (2.5 - 0);
    sedang_ = 1;
}

else if (hitung_jumlah_penghuni == 2.5)
{
    jumlah_penghuni2 = 1;
    sedang_ = 1;
}

else if (hitung_jumlah_penghuni > 2.5 && hitung_jumlah_penghuni < 5)
{
    jumlah_penghuni2 = (5 - hitung_jumlah_penghuni) / (5 - 2.5);
    sedang_ = 1;
}

else if (hitung_jumlah_penghuni >= 5)
{
    jumlah_penghuni2 = 0;
    sedang_ = 1;
}
```

```
//kondisi cukup ramai
if (hitung_jumlah_penghuni <= 2.5)
{
    jumlah_penghuni3 = 0;
    cukup_ramai = 1;
}

else if (hitung_jumlah_penghuni > 2.5 && hitung_jumlah_penghuni < 5)
{
    jumlah_penghuni3 = (hitung_jumlah_penghuni - 2.5) / (5 - 2.5);
    cukup_ramai = 1;
}

else if (hitung_jumlah_penghuni == 5)
{
    jumlah_penghuni3 = 1;
    cukup_ramai = 1;
}

else if (hitung_jumlah_penghuni > 5 && hitung_jumlah_penghuni < 7.5)
{
    jumlah_penghuni3 = (7.5 - hitung_jumlah_penghuni) / (7.5 - 5);
    cukup_ramai = 1;
}

else if (hitung_jumlah_penghuni >= 7.5)
{
    jumlah_penghuni3 = 0;
    cukup_ramai = 1;
}

//kondisi padat
if (hitung_jumlah_penghuni <= 5)
{
    jumlah_penghuni4 = 0;
    padat = 1;
}
```

```
else if (hitung_jumlah_penghuni > 5 && hitung_jumlah_penghuni < 7.5)
{
    jumlah_penghuni4 = (hitung_jumlah_penghuni - 5) / (7.5 - 5);
    padat = 1;
}

else if (hitung_jumlah_penghuni == 7.5)
{
    jumlah_penghuni4 = 1;
    padat = 1;
}

else if (hitung_jumlah_penghuni > 7.5 && hitung_jumlah_penghuni < 10)
{
    jumlah_penghuni4 = (10 - hitung_jumlah_penghuni) / (10 - 7.5);
    padat = 1;
}

else if (hitung_jumlah_penghuni >= 10)
{
    jumlah_penghuni4 = 0;
    padat = 1;
}

//jumlah sangat padat
if (hitung_jumlah_penghuni <= 7.5)
{
    jumlah_penghuni5 = 0;
    sangat_padat = 1;
}

else if (hitung_jumlah_penghuni > 7.5 && hitung_jumlah_penghuni < 10)
{
    jumlah_penghuni5 = (hitung_jumlah_penghuni - 7.5) / (10 - 7.5);
    sangat_padat = 1;
}

else if (hitung_jumlah_penghuni >= 10)
{
```



```
    jumlah_penghuni5 = 1;
    sangat_padat = 1;
}
}

void rules()
{
    //sejuk, sejuk, sepi
    if (sejuk == 1 && sejuk_dalam == 1 && sepi == 1)
    {
        rules1 = min(suhu_luar1, suhu_dalam1);
        set1 = min(rules1, jumlah_penghuni1);
        tinggi1 = set1;
    }

    //sejuk, sejuk, sedang
    if (sejuk == 1 && sejuk_dalam == 1 && sedang == 1)
    {
        rules2 = min(suhu_luar1, suhu_dalam1);
        set2 = min(rules2, jumlah_penghuni2);
        tinggi2 = set2;
    }

    //sejuk, sejuk, cukup_ramai
    if (sejuk == 1 && sejuk_dalam == 1 && cukup_ramai == 1)
    {
        rules3 = min(suhu_luar1, suhu_dalam1);
        set3 = min(rules3, jumlah_penghuni3);
        tinggi3 = set3;
    }

    //sejuk, sejuk, padat
    if (sejuk == 1 && sejuk_dalam == 1 && padat == 1)
    {
        rules4 = min(suhu_luar1, suhu_dalam1);
        set4 = min(rules4, jumlah_penghuni4);
        tinggi4 = set4;
    }
}
```

```

//sejuk, sejuk, sangat_padat
if (sejuk == 1 && sejuk_dalam == 1 && sangat_padat == 1)
{
    rules5 = min(suhu_luar1, suhu_dalam1);
    set5 = min(rules5, jumlah_penghuni5);
    tinggi5 = set5;
}

//normal, sejuk, sepi
if (normal == 1 && sejuk_dalam == 1 && sepi == 1)
{
    rules6 = min(suhu_luar2, suhu_dalam1);
    set6 = min(rules6, jumlah_penghuni1);
    tinggi6 = set6;
}

//normal, sejuk, sedang
if (normal == 1 && sejuk_dalam == 1 && sedang_ == 1)
{
    rules7 = min(suhu_luar2, suhu_dalam1);
    set7 = min(rules7, jumlah_penghuni2);
    tinggi7 = set7;
}

//normal, sejuk, cukup_ramai
if (normal == 1 && sejuk_dalam == 1 && cukup_ramai == 1)
{
    rules8 = min(suhu_luar2, suhu_dalam1);
    set8 = min(rules8, jumlah_penghuni3);
    tinggi8 = set8;
}

//normal, sejuk, padat
if (normal == 1 && sejuk_dalam == 1 && padat == 1)
{
    rules9 = min(suhu_luar2, suhu_dalam1);
    set9 = min(rules9, jumlah_penghuni4);
    sedang1 = set9;
}

```

```
//normal, sejuk, sangat_padat
if (normal == 1 && sejuk_dalam == 1 && sangat_padat == 1)
{
    rules10 = min(suhu_luar2, suhu_dalam1);
    set10 = min(rules10, jumlah_penghuni5);
    sedang2 = set10;
}

//hangat, sejuk, sepi
if (hangat == 1 && sejuk_dalam == 1 && sepi == 1)
{
    rules11 = min(suhu_luar3, suhu_dalam1);
    set11 = min(rules11, jumlah_penghunil);
    sedang3 = set11;
}

//hangat, sejuk, sedang
if (hangat == 1 && sejuk_dalam == 1 && sedang_ == 1)
{
    rules12 = min(suhu_luar3, suhu_dalam1);
    set12 = min(rules12, jumlah_penghuni2);
    sedang4 = set12;
}

//hangat, sejuk, cukup_ramai
if (hangat == 1 && sejuk_dalam == 1 && cukup_ramai == 1)
{
    rules13 = min(suhu_luar3, suhu_dalam1);
    set13 = min(rules13, jumlah_penghuni3);
    rendah1 = set13;
}

//hangat, sejuk, padat
if (hangat == 1 && sejuk_dalam == 1 && padat == 1)
{
    rules14 = min(suhu_luar3, suhu_dalam1);
```

```

set14 = min(rules14, jumlah_penghuni4);
rendah2 = set14;
}

//hangat, sejuk, sangat_padat
if (hangat == 1 && sejuk_dalam == 1 && sangat_padat == 1)
{
    rules15 = min(suhu_luar3, suhu_dalam1);
    set15 = min(rules15, jumlah_penghuni5);
    rendah3 = set15;
}

//sejuk, normal, sepi
if (sejuk == 1 && normal_dalam == 1 && sepi == 1)
{
    rules16 = min(suhu_luar1, suhu_dalam2);
    set16 = min(rules16, jumlah_penghunil);
    tinggi9 = set16;
}

//sejuk, normal, sedang
if (sejuk == 1 && normal_dalam == 1 && sedang_ == 1)
{
    rules17 = min(suhu_luar1, suhu_dalam2);
    set17 = min(rules17, jumlah_penghuni2);
    tinggi10 = set17;
}

//sejuk, normal, cukup_ramai
if (sejuk == 1 && normal_dalam == 1 && cukup_ramai == 1)
{
    rules18 = min(suhu_luar1, suhu_dalam2);
    set18 = min(rules18, jumlah_penghuni3);
    tinggi11 = set18;
}

//sejuk, normal, padat
if (sejuk == 1 && normal_dalam == 1 && padat == 1)
{

```

```
rules19 = min(suhu_luar1, suhu_dalam2);
set19 = min(rules19, jumlah_penghuni4);
sedang5 = set19;
}

//sejuk, normal, sangat_padat
if (sejuk == 1 && normal_dalam == 1 && sangat_padat == 1)
{
rules20 = min(suhu_luar1, suhu_dalam2);
set20 = min(rules20, jumlah_penghuni5);
sedang6 = set20;
}

//normal, normal, sepi
if (normal == 1 && normal_dalam == 1 && sepi == 1)
{
rules21 = min(suhu_luar2, suhu_dalam2);
set21 = min(rules21, jumlah_penghuni1);
tinggi12 = set21;
}

//normal, normal, sedang
if (normal == 1 && normal_dalam == 1 && sedang_ == 1)
{
rules22 = min(suhu_luar2, suhu_dalam2);
set22 = min(rules22, jumlah_penghuni2);
sedang7 = set22;
}

//normal, normal, cukup_ramai
if (normal == 1 && normal_dalam == 1 && cukup_ramai == 1)
{
rules23 = min(suhu_luar2, suhu_dalam2);
set23 = min(rules23, jumlah_penghuni3);
sedang8 = set23;
}
```

```
//normal, normal, padat
if (normal == 1 && normal_dalam == 1 && padat == 1)
{
    rules24 = min(suhu_luar2, suhu_dalam2);
    set24 = min(rules24, jumlah_penghuni4);
    rendah4 = set24;
}

//normal, normal, sangat_padat
if (normal == 1 && normal_dalam == 1 && sangat_padat == 1)
{
    rules25 = min(suhu_luar2, suhu_dalam2);
    set25 = min(rules25, jumlah_penghuni5);
    rendah5 = set25;
}

//hangat, normal, sepi
if (hangat == 1 && normal_dalam == 1 && sepi == 1)
{
    rules26 = min(suhu_luar3, suhu_dalam2);
    set26 = min(rules26, jumlah_penghuni1);
    sedang9 = set26;
}

//hangat, normal, sedang
if (hangat == 1 && normal_dalam == 1 && sedang_ == 1)
{
    rules27 = min(suhu_luar3, suhu_dalam2);
    set27 = min(rules27, jumlah_penghuni2);
    sedang10 = set27;
}

//hangat, normal, cukup ramai
if (hangat == 1 && normal_dalam == 1 && cukup_ramai == 1)
{
    rules28 = min(suhu_luar3, suhu_dalam2);
    set28 = min(rules28, jumlah_penghuni3);
    rendah6 = set28;
}
```

```

//hangat, normal, padat
if (hangat == 1 && normal_dalam == 1 && padat == 1)
{
    rules29 = min(suhu_luar3, suhu_dalam2);
    set29 = min(rules29, jumlah_penghuni4);
    rendah7 = set29;
}

//hangat, normal, sangat_padat
if (hangat == 1 && normal_dalam == 1 && sangat_padat == 1)
{
    rules30 = min(suhu_luar3, suhu_dalam2);
    set30 = min(rules30, jumlah_penghuni5);
    rendah8 = set30;
}

//sejuk, hangat, sepi
if (sejuk == 1 && hangat_dalam == 1 && sepi == 1)
{
    rules31 = min(suhu_luar1, suhu_dalam3);
    set31 = min(rules31, jumlah_penghunil);
    sedang11 = set31;
}

//sejuk, hangat, sedang
if (sejuk == 1 && hangat_dalam == 1 && sedang_ == 1)
{
    rules32 = min(suhu_luar1, suhu_dalam3);
    set32 = min(rules32, jumlah_penghuni2);
    sedang12 = set32;
}

//sejuk, hangat, cukup_ramai
if (sejuk == 1 && hangat_dalam == 1 && cukup_ramai == 1)
{
    rules33 = min(suhu_luar1, suhu_dalam3);

```

```

set33 = min(rules33, jumlah_penghuni3);
sedang13 = set33;
}

//sejuk, hangat, padat
if (sejuk == 1 && hangat_dalam == 1 && padat == 1)
{
rules34 = min(suhu_luar1, suhu_dalam3);
set34 = min(rules34, jumlah_penghuni4);
rendah9 = set34;
}

//sejuk, hangat, sangat_padat
if (sejuk == 1 && hangat_dalam == 1 && sangat_padat == 1)
{
rules35 = min(suhu_luar1, suhu_dalam3);
set35 = min(rules35, jumlah_penghuni5);
rendah10 = set35;
}

//normal, hangat, sepi
if (normal == 1 && hangat_dalam == 1 && sepi == 1)
{
rules36 = min(suhu_luar2, suhu_dalam3);
set36 = min(rules36, jumlah_penghunil);
sedang14 = set36;
}

//normal, hangat, sedang
if (normal == 1 && hangat_dalam == 1 && sedang_ == 1)
{
rules37 = min(suhu_luar2, suhu_dalam3);
set37 = min(rules37, jumlah_penghuni2);
sedang15 = set37;
}

//normal, hangat, cukup_ramai
if (normal == 1 && hangat_dalam == 1 && cukup_ramai == 1)
{

```



```

rules38 = min(suhu_luar2, suhu_dalam3);
set38 = min(rules38, jumlah_penghuni3);
rendah11 = set38;
}

//normal, hangat, padat
if (normal == 1 && hangat_dalam == 1 && padat == 1)
{
rules39 = min(suhu_luar2, suhu_dalam3);
set39 = min(rules39, jumlah_penghuni4);
rendah12 = set39;
}

//normal, hangat, sangat_padat
if (normal == 1 && hangat_dalam == 1 && sangat_padat == 1)
{
rules40 = min(suhu_luar2, suhu_dalam3);
set40 = min(rules40, jumlah_penghuni5);
rendah13 = set40;
}

//hangat, hangat, sepi
if (hangat == 1 && hangat_dalam == 1 && sepi == 1)
{
rules41 = min(suhu_luar3, suhu_dalam3);
set41 = min(rules41, jumlah_penghuni1);
sedang16 = set41;
}

//hangat, hangat, sedang
if (hangat == 1 && hangat_dalam == 1 && sedang_ == 1)
{
rules42 = min(suhu_luar3, suhu_dalam3);
set42 = min(rules42, jumlah_penghuni2);
rendah14 = set42;
}

//hangat, hangat, cukup_ramai
if (hangat == 1 && hangat_dalam == 1 && cukup_ramai == 1)

```

```

{
    rules43 = min(suhu_luar3, suhu_dalam3);
    set43 = min(rules43, jumlah_penghuni3);
    rendah15 = set43;
}

//hangat, hangat, padat
if (hangat == 1 && hangat_dalam == 1 && padat == 1)
{
    rules44 = min(suhu_luar3, suhu_dalam3);
    set44 = min(rules44, jumlah_penghuni4);
    rendah16 = set44;
}

//hangat, hangat, sangat_padat
if (hangat == 1 && hangat_dalam == 1 && sangat_padat == 1)
{
    rules45 = min(suhu_luar3, suhu_dalam3);
    set45 = min(rules45, jumlah_penghuni5);
    rendah17 = set45;
}
}

void defuzzyfikasi()
{
    //tinggi
    maximum_tinggi1 = max(tinggi1, tinggi2);
    maximum_tinggi2 = max(tinggi3, tinggi4);
    maximum_tinggi3 = max(tinggi5, tinggi6);
    maximum_tinggi4 = max(tinggi7, tinggi8);
    maximum_tinggi5 = max(tinggi9, tinggi10);
    maximum_tinggi6 = max(tinggi11, tinggi12);

    maximum_tinggi7 = max(maximum_tinggi1, maximum_tinggi2);
    maximum_tinggi8 = max(maximum_tinggi3, maximum_tinggi4);
    maximum_tinggi9 = max(maximum_tinggi5, maximum_rendah6);

    maximum_tinggi10 = max(maximum_tinggi7, maximum_tinggi8);
    maximum_tinggi11 = max(maximum_tinggi9, maximum_tinggi10);

```

```

//Sedang
maximum_sedang1 = max(sedang1, sedang2);
maximum_sedang2 = max(sedang3, sedang4);
maximum_sedang3 = max(sedang5, sedang6);
maximum_sedang4 = max(sedang7, sedang8);
maximum_sedang5 = max(sedang9, sedang10);
maximum_sedang6 = max(sedang11, sedang12);
maximum_sedang7 = max(sedang13, sedang14);
maximum_sedang8 = max(sedang15, sedang16);

maximum_sedang9 = max(maximum_sedang1, maximum_sedang2);
maximum_sedang10 = max(maximum_sedang3, maximum_sedang4);
maximum_sedang11 = max(maximum_sedang5, maximum_sedang6);
maximum_sedang12 = max(maximum_sedang7, maximum_sedang8);
maximum_sedang13 = max(maximum_sedang9, maximum_sedang10);
maximum_sedang14 = max(maximum_sedang11, maximum_sedang12);
maximum_sedang15 = max(maximum_sedang14, maximum_sedang13);

//rendah
maximum_rendah1 = max(rendah1, rendah2);
maximum_rendah2 = max(rendah3, rendah4);
maximum_rendah3 = max(rendah5, rendah6);
maximum_rendah4 = max(rendah7, rendah8);
maximum_rendah5 = max(rendah9, rendah10);
maximum_rendah6 = max(rendah11, rendah12);
maximum_rendah7 = max(rendah13, rendah14);
maximum_rendah8 = max(rendah15, rendah16);
maximum_rendah9 = max(maximum_rendah1, maximum_rendah2);
maximum_rendah10 = max(maximum_rendah3, maximum_rendah4);
maximum_rendah11 = max(maximum_rendah5, maximum_rendah6);
maximum_rendah12 = max(maximum_rendah7, maximum_rendah8);
maximum_rendah13 = max(maximum_rendah9, maximum_rendah10);
maximum_rendah14 = max(maximum_rendah11, maximum_rendah12);
maximum_rendah15 = max(maximum_rendah13, maximum_rendah14);
maximum_rendah16 = max(maximum_rendah15, renda
}

```

```

void hasil()
{
    hasil = ceil (((maximum_rendah16 * output_1) + (maximum_sedang15 *
output_2) + (maximum_tinggi11 * output_3)) / (maximum_rendah16 +
maximum_sedang15 + maximum_tinggi11));
}

```

B. Slave Program

```

#include <EEPROM.h>

#define on_off 3
#define setting_up 4
#define setting_down 5

int terima_data;

int in = 0;
int looping;

byte alamat_eeprom;

void setup() {
    // put your setup code here, to run once:
    Serial.begin(9600);

    pinMode(on_off, OUTPUT);
    pinMode(setting_up, OUTPUT);
    pinMode(setting_down, OUTPUT);

    //SET NILAI DEFAULT
    EEPROM.write(0, 25);
}

void loop() {
    // put your main code here, to run repeatedly:
    alamat_eeprom = EEPROM.read(0);

    if (Serial.available())

```

```
{
  terima_data = Serial.read();

  if (terima_data < alamat_eeprom)
  { in = 0;

    if (in == 0)
    {
      for (looping = terima_data; looping < alamat_eeprom; looping++)
      {
        digitalWrite(setting_up, HIGH);
        delay(300);
        digitalWrite(setting_up, LOW);
        delay(300);

        if (looping == alamat_eeprom)
        {
          in = 1;
          EEPROM.write(0, looping);
        }
      }
    }
  }

  if (terima_data > alamat_eeprom)
  { in = 0;

    if (in == 0)
    {
      for (looping = terima_data; looping > alamat_eeprom; looping--)
      {
        digitalWrite(setting_down, HIGH);
        delay(300);
        digitalWrite(setting_down, LOW);
        delay(300);

        if (looping == alamat_eeprom)
        {
          in = 1;

```

```
EEPROM.write(0, looping);  
}  
}  
}  
}  
}  
}
```

RIWAYAT HIDUP



Reza Fahlevi, lahir di Jakarta pada tanggal 10 Desember 1995 dari pasangan Bapak Cecep Suganda dan Ibu Arfah sebagai anak kedua dari dua bersaudara. Memiliki nama panggilan Reza yang bertempat tinggal di Jalan Gongseng Raya RT.06/09 Kel. Baru, Kec. Pasar Rebo, Jakarta Timur. Riwayat pendidikan formal penulis dimulai dari pendidikan Sekolah Dasar di SD Negeri 11 Petang pada tahun 2001 dan lulus pada tahun 2007. Kemudian melanjutkan ke jenjang Sekolah Menengah Pertama di SMP Negeri 179 Jakarta pada tahun 2007 dan lulus pada tahun 2010. Lalu penulis melanjutkan ke jenjang Sekolah Menengah Atas di SMAN 88 Jakarta pada tahun 2010 dan lulus pada tahun 2013. Hingga akhirnya setelah tamat Sekolah Menengah Atas, penulis melanjutkan ke jenjang Perguruan Tinggi pada tahun 2013 di Universitas Negeri Jakarta, Fakultas Teknik, Program Studi Pendidikan Teknik Elektro, Konsentrasi Peminatan Otomasi Industri.

Selama perkuliahan, penulis juga aktif di beberapa organisasi kemahasiswaan di dalam kampus. Di dalam kampus meliputi HMJ Teknik Elektro UNJ, Forum Studi Islam Al-Biruni, dan Klub Robotik FT UNJ.