

IMPLEMENTASI PENGKONDISIAN KIPAS DAN LAMPU OTOMATIS MENGGUNAKAN LOGIKA FUZZY

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Randy Maulana
NIM: 135150301111029



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

IMPLEMENTASI PENGKONDISIAN KIPAS DAN LAMPU OTOMATIS MENGGUNAKAN LOGIKA FUZZY

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Diajukan untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Komputer

Disusun Oleh :

Randy Maulana

NIM: 135150301111029

Skripsi ini telah diuji dan dinyatakan lulus pada

6 Juni 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Mochammad Hannats Hanafi Ichsan, S.ST, M.T

NIK: 201405 881229 1 001

Gembong Edhi Setyawan, S.T, M.T

NIK: 201208 761201 1 001

Mengetahui

Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T, M.T, Ph.D

NIP: 19710518 200312 1 001

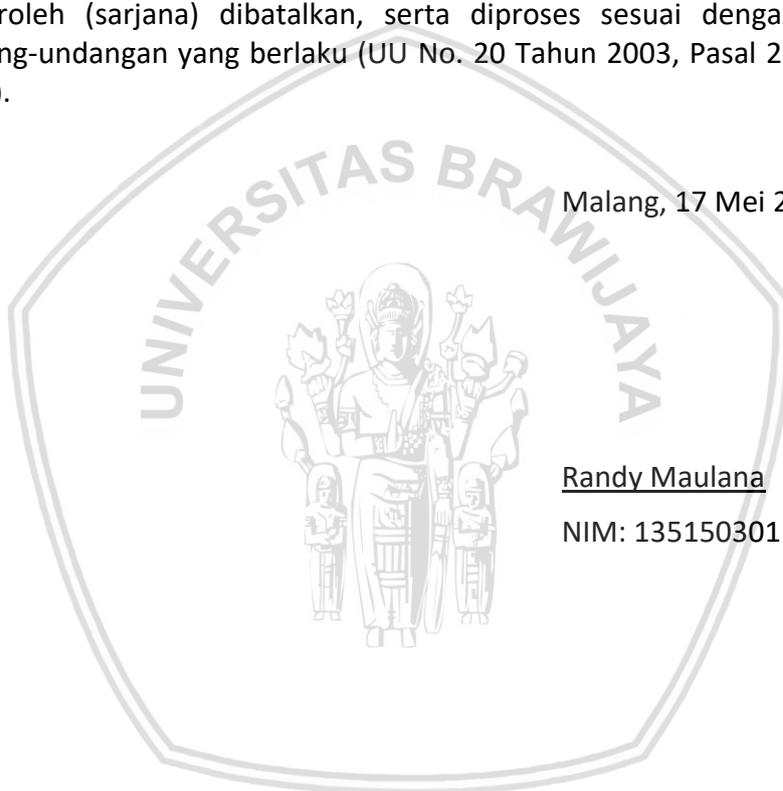


PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 17 Mei 2018



Randy Maulana

NIM: 135150301111029

KATA PENGANTAR

Puji syukur kehadirat Tuhan Yang Maha Esa yang telah melimpahkan Rahmat, dan Berkatnya-Nya sehingga laporan skripsi yang berjudul “Implementasi Pengkondisian Kipas dan Lampu Otomatis Menggunakan Logika Fuzzy” ini dapat terselesaikan.

Banyak kesulitan dan hambatan yang dialami oleh penulis dalam pembuatan skripsi ini, tetapi semua itu dapat diatasi dengan baik berkat dukungan dan bantuan dari berbagai pihak. Oleh karena itu, penulis ingin menyampaikan rasa hormat dan terima kasih kepada:

1. Kedua Orang Tua dan seluruh keluarga besar atas segala nasehat, kasih sayang dan kesabaran dalam membesarkan dan mendidik penulis, serta senantiasa tiada hentinya memberikan doa dan semangat demi terselesainya skripsi ini.
2. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D. selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya Malang.
3. Bapak Heru Nurwarsito, Ir., M. Kom. selaku Wakil Ketua I Bidang Akademik Fakultas Ilmu Komputer Universitas Brawijaya Malang.
4. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya Malang.
5. Bapak Sabriansyah Rizqika Akbar, S.T, M.Eng. selaku Ketua Program Studi Teknik Komputer Universitas Brawijaya Malang.
6. Bapak Mochammad Hannats Hanafi Ichsan, S.ST, M.T, dan Bapak Gembong Edhi Setyawan, S.T, M.T. selaku dosen pembimbing skripsi penulis yang dengan sabar membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini.
7. Seluruh civitas akademik Fakultas Ilmu Komputer Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama penyelesaian skripsi ini.
8. Teman-teman Teknik Komputer angkatan 2013 yang selalu mendukung dan berbagi ilmu dari awal perkuliahan sampai tahap akhir penyelesaian skripsi dan semua pihak yang telah membantu terselesainya skripsi ini yang tidak dapat penulis sebutkan satu per satu.
9. Teman-teman Capoeira Senzala Malang yang selalu mendukung dan memberikan semangat dalam pengerjaan skripsi ini yang tidak dapat penulis sebutkan satu per satu.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih jauh dari sempurna, sehingga penulis mengharapkan adanya segala saran dan kritik yang membangun. Akhir kata penulis berharap skripsi ini dapat membawa manfaat bagi semua pihak yang menggunakannya.

Malang, 17 Mei 2018

Penulis

randy.m95@gmail.com



ABSTRAK

Kebutuhan energi listrik di Indonesia semakin meningkat bersamaan dengan pertumbuhan ekonomi dan industri nasional serta pertumbuhan penduduk setiap tahunnya. Peningkatan ini terjadi karena peralatan-peralatan elektronik seperti pendingin ruangan dibiarkan menyala meskipun tidak ada orang pada ruangan, contoh yang sama pada TV dan yang dibiarkan menyala terus pada sebuah ruangan. Kebiasaan umum seperti ini disebabkan karena prinsip dari peralatan-peralatan elektronik ini menggunakan prinsip *on-off*. Untuk mengatasi masalah-masalah tersebut pada penelitian ini akan mengimplementasikan suatu sistem pengkondisian kipas dan lampu dengan menggunakan logika *Fuzzy*. Sistem akan menggunakan beberapa perangkat keras seperti mikrokontroler Arduino Mega dengan sensor *input* dari LDR sebagai sensor cahaya, DHT11 sebagai sensor suhu, dan dua sensor ultrasonik sebagai sensor gerak dan jarak, dan hasil *output* akan menggunakan DC motor sebagai kipas, dan LED sebagai lampu. Logika *Fuzzy* pada sistem akan menggunakan bantuan *Embedded Fuzzy Logic Library* (eFLL) digunakan untuk memudahkan implementasi proses fuzzifikasi yaitu menentukan fungsi keanggotaan *input* dan *output*, proses inferensi pembentukan aturan *Fuzzy* dan proses defuzzifikasi pengkonversian nilai *input* menjadi nilai tegas. Tingkat akurasi dan keberhasilan logika *Fuzzy* pada sistem akan diuji dengan membandingkan nilai *output* sistem pada alat dengan nilai *output* sistem pada MATLAB. Rata-rata akurasi yang didapat untuk *output* kipas mencapai 98,77% dan *output* lampu mencapai 92,54%.

Kata kunci: logika Fuzzy, otomatis, kipas, lampu, ruangan

ABSTRACT

Electrical energy needs in Indonesia is increasing along with economic growth and national industry and population growth every year. This increase occurs because electronic appliances such as air conditioners are left on, even though no one is in the room, the same example on the TV and left lit continuously in a room. This common practice is due to the principle of electronic devices using the on-off principle. To overcome these problems in this study will implement a system of fan and lamp conditioning using Fuzzy logic. The system will use some hardware such as Arduino Mega microcontroller with LDR input sensor as light sensor, DHT11 as temperature sensor, and two ultrasonic sensors as motion and distance sensors, and output will use DC motor as fan, and LED as lamp. Fuzzy logic on the system will use the help of Embedded Fuzzy Logic Library (eFLL) is used to facilitate the implementation of fuzzification process that determines the membership function of input and output, the inference process of Fuzzy rule formation and defuzzification process of converting the input value into firm value. The level of accuracy and success of Fuzzy logic on the system will be tested by comparing the value of system output on the tool with the system output value in MATLAB. The average accuracy obtained for fan output reached 98.77% and the lamp output reached 92.54%.

Keywords: *Fuzzy logic, automatic, fan, lamp, room*

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xiii
DAFTAR LAMPIRAN	xv
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan	2
1.4 Batasan masalah	3
1.5 Sistematika pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Kajian Pustaka	5
2.2 Dasar Teori.....	6
2.2.1 Logika <i>Fuzzy</i>	6
2.2.2 Arduino Mega.....	10
2.2.3 Sensor Ultrasonik HC-SR04	12
2.2.4 Sensor DHT11	12
2.2.5 Sensor <i>Light Dependant Resistor</i> (LDR)	13
2.2.6 <i>Driver Motor</i> L298N	14
BAB 3 METODOLOGI	15
3.1 Studi Literatur	16
3.2 Analisis Kebutuhan	16
3.3 Perancangan dan Implementasi	16
3.4 Pengujian dan Analisis	17
3.5 Kesimpulan dan Saran	17



BAB 4 ANALISIS KEBUTUHAN	18
4.1 Kebutuhan Pengguna.....	18
4.2 Kebutuhan Sistem.....	19
4.2.1 Kebutuhan Perangkat Keras.....	19
4.2.2 Kebutuhan Perangkat Lunak	21
4.3 Kebutuhan Fungsional	22
4.4 Kebutuhan Non Fungsional.....	22
4.4.1 Karakteristik Pengguna	22
4.4.2 Lingkungan Pengguna	22
4.4.3 Batasan Desain Sistem	23
BAB 5 PERANCANGAN DAN IMPLEMENTASI	24
5.1 Perancangan Sistem.....	24
5.2 Perancangan Perangkat Sistem	25
5.2.1 Perancangan Input Sensor Gerak.....	25
5.2.2 Perancangan Input Sensor Jarak	26
5.2.3 Perancangan Input Sensor Suhu	27
5.2.4 Perancangan Input Sensor Cahaya	28
5.2.5 Perancangan Output Kipas dan Lampu.....	29
5.2.6 Perancangan Alat Prototipe Sistem	30
5.3 Fuzzifikasi	31
5.3.1 Perancangan Fungsi Keanggotaan Input.....	31
5.3.2 Implementasi Fungsi Keanggotaan Input pada Arduino.....	36
5.4 Inferensi	38
5.4.1 Pembentukan <i>Rule Fuzzy</i>	38
5.4.2 Implementasi Inferensi pada Arduino	39
5.5 Defuzzifikasi	40
5.5.1 Perancangan Fungsi Keanggotaan Output.....	41
5.5.2 Implementasi Defuzzifikasi pada Arduino	44
5.6 Implementasi Perangkat Sistem dan MATLAB	45
5.6.1 Implementasi Sensor Gerak.....	45
5.6.2 Implementasi Sensor Jarak	46
5.6.3 Implementasi Sensor Suhu	47



5.6.4 Implementasi Sensor Cahaya	48
5.6.5 Implementasi Kipas dan Lampu	49
5.6.6 Implementasi Alat Prototipe Sistem	49
5.6.7 Implementasi MATLAB.....	50
BAB 6 PENGUJIAN DAN ANALISIS.....	52
6.1 Pengujian Sensor Ultrasonik Gerak dan Jarak	52
6.2 Pengujian Sensor DHT11.....	54
6.3 Pengujian Sensor LDR	55
6.4 Pengujian Kesesuaian <i>Output</i>	56
6.5 Pengujian Logika <i>Fuzzy</i> dengan MATLAB.....	57
BAB 7 KESIMPULAN DAN SARAN	60
7.1 Kesimpulan.....	60
7.2 Saran	60
DAFTAR PUSTAKA.....	61
LAMPIRAN A DAFTAR <i>RULE FUZZY</i>.....	63
LAMPIRAN B DATA SURVEY	68



DAFTAR TABEL

Tabel 2.1 Spesifikasi Arduino Mega	11
Tabel 2.2 Spesifikasi sensor ultrasonik HC-SR04.....	12
Tabel 2.3 Spesifikasi sensor DHT11.....	13
Tabel 5.1 Hubungan pin gerak dengan Arduino Mega	25
Tabel 5.2 Hubungan pin sensor jarak dengan Arduino Mega.....	26
Tabel 5.3 Hubungan pin sensor suhu dengan Arduino Mega.....	27
Tabel 5.4 Hubungan pin sensor cahaya dengan Arduino Mega	28
Tabel 5.5 Hubungan pin <i>driver motor</i> dan pin LED dengan Arduino Mega.....	29
Tabel 5.6 <i>Domain range input</i> gerak.....	32
Tabel 5.7 <i>Domain range input</i> jarak.....	33
Tabel 5.8 <i>Domain range input</i> suhu	34
Tabel 5.9 <i>Domain range input</i> cahaya	35
Tabel 5.10 Kode sumber fungsi <i>FuzzySet input</i>	37
Tabel 5.11 Kode sumber fungsi <i>FuzzyInput object</i>	37
Tabel 5.12 Kode sumber fungsi fuzzifikasi	38
Tabel 5.13 Potongan Daftar <i>Rule Fuzzy</i>	39
Tabel 5.14 Potongan kode sumber fungsi <i>FuzzyRuleAntecedent object</i>	40
Tabel 5.15 Potongan kode sumber fungsi <i>FuzzyRuleConsequent</i>	40
Tabel 5.16 <i>Domain range output</i> kipas.....	42
Tabel 5.17 <i>Domain range output</i> lampu	43
Tabel 5.18 Potongan kode sumber fungsi keanggotaan fuzzy output	44
Tabel 5.19 Potongan kode sumber fungsi <i>FuzzyRule object</i>	45
Tabel 5.20 Potongan kode sumber defuzzifikasi	45
Tabel 5.21 Kode sumber sensor gerak.....	46
Tabel 5.22 Kode sumber sensor jarak.....	47
Tabel 5.23 Kode sumber <i>input</i> suhu	48
Tabel 5.24 Kode program <i>input</i> cahaya	48
Tabel 5.25 Kode sumber ouput kipas dan lampu	49
Tabel 6.1 Hasil pengujian akurasi sensor ultrasonik gerak dan jarak	52
Tabel 6.2 Hasil pengujian akurasi sensor DHT11	54

Tabel 6.3 Hasil pengujian akurasi sensor LDR..... 55
Tabel 6.4 Hasil pengujian kesesuaian *output* sistem 57
Tabel 6.5 Hasil pengujian akurasi *output* sistem dengan *output* MATLAB..... 58



DAFTAR GAMBAR

Gambar 2.1 Representasi linear naik	7
Gambar 2.2 Representasi linear turun.....	7
Gambar 2.3 Representasi kurva segitiga	8
Gambar 2.4 Representasi kurva trapesium	8
Gambar 2.5 Bentuk Fisik Arduino ATmega1280	11
Gambar 2.6 Bentuk fisik sensor ultrasonik	12
Gambar 2.7 Bentuk Fisik DHT11	13
Gambar 2.8 Bentuk fisik sensor LDR	13
Gambar 2.9 Bentuk fisik <i>driver motor</i> L298N	14
Gambar 3.1 Diagram Alir penelitian	15
Gambar 3.2 Diagram perancangan <i>input</i> , <i>proses</i> , dan <i>output</i>	16
Gambar 4.1 Diagram Analisis Kebutuhan	18
Gambar 4.2 Tampilan dari aplikasi Arduino IDE	21
Gambar 5.1 Diagram alir perancangan sistem.....	24
Gambar 5.2 Rangkaian Sensor Gerak.....	26
Gambar 5.3 Rangkaian Sensor Jarak.....	27
Gambar 5.4 Rangkaian Sensor Suhu	28
Gambar 5.5 Rangkaian Sensor Cahaya	29
Gambar 5.6 Rangkaian <i>output</i> kipas dan lampu	30
Gambar 5.7 Perancangan alat prototipe sistem	30
Gambar 5.8 Diagram alir proses fuzzifikasi.....	31
Gambar 5.9 Kurva fungsi keanggotaan <i>input</i> gerak.....	32
Gambar 5.10 Kurva fungsi keanggotaan <i>input</i> jarak.....	33
Gambar 5.11 Kurva fungsi keanggotaan <i>input</i> suhu.....	34
Gambar 5.12 Kurva fungsi keanggotaan <i>input</i> cahaya	35
Gambar 5.13 Diagram Alir Inferensi	38
Gambar 5.14 Diagram alir defuzzifikasi	41
Gambar 5.15 Kurva fungsi keanggotaan <i>output</i> kipas.....	41
Gambar 5.16 Kurva fungsi keanggotaan <i>output</i> lampu	43
Gambar 5.17 Implementasi <i>input</i> sensor gerak.....	46

Gambar 5.18 Implementasi *input* sensor jarak..... 47

Gambar 5.19 Implementasi *input* sensor suhu..... 47

Gambar 5.20 Implementasi *input* sensor cahaya 48

Gambar 5.21 Implementasi *output* kipas dan lampu 49

Gambar 5.22 Implementasi alat prototipe sistem..... 50

Gambar 5.23 *Fuzzy Logic Toolbox* pada MATLAB 50

Gambar 5.24 *Membership Function Editor* untuk fungsi keanggotaan..... 51

Gambar 5.25 *Rule Editor* untuk pemodelan aturan *fuzzy*..... 51

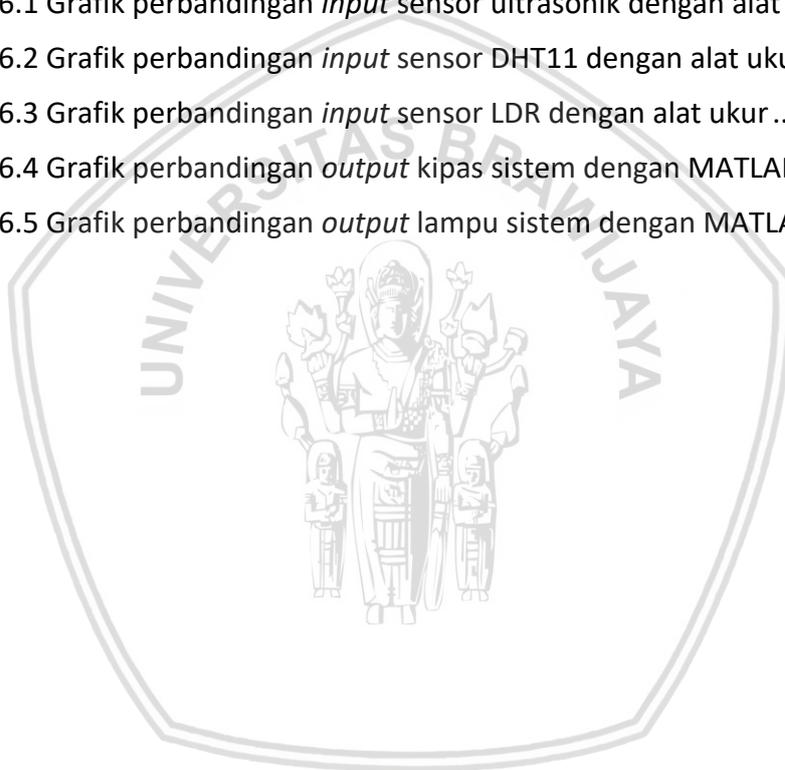
Gambar 6.1 Grafik perbandingan *input* sensor ultrasonik dengan alat ukur 53

Gambar 6.2 Grafik perbandingan *input* sensor DHT11 dengan alat ukur 55

Gambar 6.3 Grafik perbandingan *input* sensor LDR dengan alat ukur 56

Gambar 6.4 Grafik perbandingan *output* kipas sistem dengan MATLAB 59

Gambar 6.5 Grafik perbandingan *output* lampu sistem dengan MATLAB 59



BAB 1 PENDAHULUAN

1.1 Latar belakang

Kebutuhan energi listrik di Indonesia semakin meningkat bersamaan dengan pertumbuhan ekonomi dan industri nasional serta pertumbuhan penduduk. Kebutuhan energi listrik tersebut dibagi menjadi beberapa sektor diantaranya sektor industri, sektor rumah tangga, sektor transportasi, dan sektor komersial. Sektor rumah tangga menjadi salah satu sektor yang paling mendominasi akan kebutuhan energi listrik pada tahun 2013, karena dengan adanya perkembangan teknologi peralatan rumah tangga, seperti TV, pendingin ruangan, kulkas, mesin cuci, dan lain-lain yang saat ini menjadi kebutuhan utama. Di kota-kota besar kebutuhan akan penggunaan pendingin ruangan semakin meningkat. Akibatnya kebutuhan listrik pada sektor rumah tangga diperkirakan akan meningkat tajam. Pemanfaatan listrik per rumah tangga pada tahun 2013 adalah sebesar 3,057 kWh/RT dan diperkirakan meningkat menjadi 24,448 kWh/RT di tahun 2050 (Badan Pengkajian dan Penerapan Teknologi, 2015). Peningkatan ini terjadi karena peralatan-peralatan elektronik seperti pendingin ruangan dibiarkan menyala meskipun tidak ada orang pada ruangan, contoh yang sama pada TV dan yang dibiarkan menyala terus pada sebuah ruangan. Kebiasaan umum seperti ini disebabkan karena prinsip dari peralatan-peralatan elektronik ini menggunakan prinsip *on-off*. Kondisi ini dapat mengakibatkan penggunaan energi listrik yang tidak efisien dan berkurangnya masa pakai pada peralatan-peralatan elektronik tersebut (Tarigan, et al., 2015). Dengan adanya penggunaan energi listrik yang seperti ini maka diperlukan suatu sistem kendali atau pengkondisian peralatan-peralatan elektronik yang akan mengatasi masalah tersebut.

Penelitian-penelitian sebelumnya yang telah dilakukan oleh Tarigan (2013) telah membuat pengendalian alat pendingin ruangan secara otomatis yang dapat membantu dalam penghematan listrik. Penelitian tersebut menggunakan mikrokontroler ATmega 8535 dan dua jenis sensor input yaitu LM35 sebagai sensor suhu dan PIR (*Passive Infrared*) sebagai sensor gerak. Sistem menggunakan metode *fuzzy logic* untuk melakukan pengambilan keputusan yang selanjutnya akan diproses oleh mikrokontroler. Namun hasil pada penelitian tersebut masih dalam bentuk simulasi pendingin ruangan yang dapat mendeteksi gerakan manusia dan suhu dalam sebuah ruangan.

Penelitian berikutnya dari Mulia (2017) telah dibuat sebuah sistem pengkondisian nyala AC dan Lampu secara otomatis dengan menggunakan logika *Fuzzy Sugeno*. Adapun beberapa sensor input yang digunakan diantaranya adalah LDR sebagai sensor cahaya, DHT11 sebagai sensor suhu dan kelembaban, PIR sebagai sensor gerak, dan Mikrokontroler Arduino Uno untuk menjalankan logika fuzzy. Namun dari sisi *output* AC dan Lampu masih menggunakan angka yang ditampilkan pada LCD. Maka dari itu pada penelitian ini akan digunakan output dengan menggunakan perangkat keras kipas dan lampu.

Berdasarkan latar belakang yang telah disampaikan, pada penelitian ini penulis akan melakukan pengkondisian kipas dan lampu dengan menggunakan logika *Fuzzy*. Alasannya, logika *fuzzy* merupakan komponen dari soft computing dan telah banyak diaplikasikan di berbagai bidang kehidupan. Salah satu aplikasi terpentingnya adalah untuk membantu manusia dalam melakukan pengambilan keputusan, karena banyaknya kondisi yang menuntut adanya keputusan yang tidak hanya bisa dijawab dengan 'Ya' atau 'Tidak' (Kusumadewi & Purnomo, 2013). Logika *Fuzzy* menjadi sangat populer dalam bidang kontrol, karena pada umumnya proses dari kontrol logika *Fuzzy* sangat sederhana untuk merealisasikan atau mewujudkan strategi kontrol manusia, dimana kontrol konvensional sangat bergantung pada model matematis yang sesuai (Oltean & Ivanciu, 2017). Alasan lain mengapa digunakannya logika *fuzzy* untuk pengkondisian kipas dan lampu pada penelitian ini yaitu logika *fuzzy* sangat fleksibel, memiliki toleransi terhadap data-data yang tidak tepat, mampu memodelkan fungsi-fungsi *nonlinear* yang kompleks, dan dalam pengaplikasiannya daya guna dan pengendaliannya dianggap lebih baik daripada teknik kendali lain, mudah diperbaiki, usaha dan dana yang dibutuhkan lebih kecil (Saelan, 2009).

Penelitian ini akan mengimplementasikan sistem pengkondisian kipas dan lampu dengan menggunakan beberapa perangkat keras seperti mikrokontroler Arduino Mega dengan sensor *input* dari LDR sebagai sensor cahaya, DHT11 sebagai sensor suhu, dan dua sensor ultrasonik sebagai sensor gerak dan jarak, dan hasil *output* akan menggunakan DC motor sebagai kipas, dan LED sebagai lampu. Sehingga akan didapatkan hasil *output* yang dapat menyesuaikan keadaan suhu dan cahaya yang sesungguhnya pada sebuah ruangan.

1.2 Rumusan masalah

Berdasarkan latar belakang yang telah disampaikan di atas, dapat dirumuskan beberapa rumusan masalah pada penelitian ini adalah sebagai berikut:

1. Bagaimana rancangan pengkondisian kipas dan lampu otomatis dengan menggunakan logika *fuzzy*?
2. Bagaimana implementasi pengkondisian kipas dan lampu otomatis menggunakan logika *fuzzy* ke Arduino?
3. Bagaimana hasil pengujian tingkat akurasi sensor *input*, *output* dan metode logika *fuzzy* dalam pengkondisian kipas dan lampu otomatis tersebut?

1.3 Tujuan

1. Merancang dan membuat sistem pengkondisian kipas dan lampu otomatis dengan menggunakan logika *fuzzy*.
2. Mengimplementasikan logika *fuzzy* pada Arduino sebagai sistem kendali otomatis kipas dan lampu.
3. Menguji tingkat akurasi logika *fuzzy* sistem dalam pengkondisian kipas dan lampu otomatis.

1.4 Batasan masalah

1. Menggunakan mikrokontroler Arduino Mega.
2. *Input* menggunakan sensor DHT11 untuk mendeteksi suhu, sensor LDR untuk mendeteksi cahaya, sensor ultrasonik untuk mendeteksi gerak dan jarak seseorang pada ruangan.
3. *Output* menggunakan alat DC motor sebagai kipas dan led sebagai lampu.
4. Logika *fuzzy* diimplementasikan pada Arduino Mega.
5. Sistem akan di implementasikan menggunakan perangkat prototype.

1.5 Sistematika pembahasan

Berikut merupakan sistematika pembahasan dalam penyusunan laporan ini:

BAB 1 PENDAHULUAN

Pada bab ini akan menjelaskan latar belakang yang menjadi alasan penelitian yang dilakukan, rumusan masalah yang berisi pertanyaan penelitian, tujuan yang akan dicapai pada penelitian, batasan masalah dari penelitian dan sistematika dari pembahasan dan penulisan dari penelitian “Impelementasi Pengkondisian Kipas dan Lampu Otomatis Menggunakan Logika Fuzzy”.

BAB 2 LANDASAN KEPUSTAKAAN

Bab ini berisi kajian tentang penelitian sebelumnya yang relevan dan memaparkan teori-teori dan konsep-konsep yang didapat dari sumber-sumber yang relevan seperti Logika Fuzzy, Arduino Mega dan sensor-sensor apa saja yang akan digunakan pada penelitian sebagai panduan dalam penelitian serta penyusunan laporan.

BAB 3 METODOLOGI

Bab ini membahas metode, prosedur dan langkah-langkah yang digunakan pada penelitian ini. Di sini berisi pemaparan langkah kerja yang terdiri atas studi literatur, analisis kebutuhan mulai dari kebutuhan perangkat keras dan perangkat lunak, perancangan, implementasi, pengujian dan analisis, dan kesimpulan dan saran dari peneliti.

BAB 4 ANALISIS KEBUTUHAN

Bab ini akan membahas gambaran umum dari sistem yang akan dibuat dan analisis kebutuhan yang meliputi kebutuhan perangkat keras, perangkat lunak, kebutuhan fungsional, dan kebutuhan non-fungsional dari sistem.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

Bab ini menguraikan perancangan sistem yang terdiri dari perancangan logika *fuzzy*, perancangan perangkat keras, dan

perancangan MATLAB, yang selanjutnya akan dilakukan implementasi alat dan prototipe dari perancangan yang telah dibuat.

BAB 6 PENGUJIAN DAN ANALISIS

Bab ini memuat hasil pengujian dan analisis terhadap sistem yang telah dibangun beserta hasil pengujian dari masing-masing sensor *input*, pengujian kesesuaian *output*, dan pengujian logika *fuzzy* dengan MATLAB.

BAB 7 KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan atas penelitian yang telah dikerjakan serta memberi saran untuk pengembangan lebih lanjut dari sistem yang telah dibangun.



BAB 2 LANDASAN KEPUSTAKAAN

Pada bab ini akan menjelaskan berbagai kajian pustaka sebagai referensi dari penelitian-penelitian terdahulu yang sudah dilakukan oleh peneliti sebelumnya dan dasar teori pendukung dalam penelitian yang akan dilakukan, seperti Logika Fuzzy, Sensor Ultrasonik, Sensor DHT11, Sensor LDR, dan Driver Motor L298N.

2.1 Kajian Pustaka

Pada penelitian sebelumnya dari Tarigan (2015) telah melakukan analisis otomatisasi pada sebuah bangunan menggunakan logika fuzzy Mamdani. Penelitian ini menggunakan 3 *input* variabel Fuzzy diantaranya sensor cahaya, sensor suhu, dan sensor pergerakan untuk mengetahui keberadaan seseorang dalam ruangan. Kemudian, kontroller logika Fuzzy digunakan sebagai pengatur AC, TV, dan pencahayaan/lampu. Pada hasil dan analisis didapatkan kelebihan dari logika fuzzy yang digunakan mampu melakukan kinerja sistem yang optimal namun kekurangannya penelitian masih menggunakan data angka untuk menampilkan persentasi kinerja output sistem, tidak menggunakan media perangkat keras secara nyata.

Penelitian yang kedua dari Zaki (2016) telah menangani sebuah masalah pengaturan otomatisasi terang lampu pada smart home. Pada penelitian ini metode yang digunakan adalah algoritma forward chaining dan prototype menggunakan lampu LED, sensor intensitas cahaya BH1750 sebagai pendeteksi cahaya di dalam dan di luar ruangan, passive infrared HC-SR501 sebagai pendeteksi pergerakan manusia di dalam ruangan, dan mikrokontroler Arduino Uno R3. Hasil yang diperoleh secara keseluruhan bahwa sistem dapat menghemat energi listrik $\pm 0,035$ Watt dari pemakaian 0,059 Watt. Namun pada penelitian ini sistem hanya menggunakan satu output yaitu lampu.

Penelitian selanjutnya dari Mulia (2017) telah membuat sebuah sistem pengkondisian nyala AC dan lampu otomatis dengan menggunakan logika Fuzzy Sugeno. Sistem yang dibuat menggunakan *input* cahaya menggunakan sensor LDR, suhu dan kelembaban menggunakan sensor DHT11, dan pergerakan menggunakan sensor PIR HC-SR501 dan menggunakan Arduino Uno sebagai mikrokontroler dan hasil atau menggunakan sebuah LCD yang akan menampilkan angka. Hasil dari implementasi di dapatkan dari tingkat keberhasilan logika Fuzzy dengan cara membandingkan nilai *output* sistem dengan MATLAB. Dan di dapatkan rata-rata akurasi AC sebesar 97,21%, dan lampu sebesar 93,75%. Kelebihan dari penelitian ini bahwa sistem mampu melakukan otomatisasi berdasarkan input yang diperoleh dari sensor, namun output yang digunakan pada penelitian ini hanya menggunakan LCD untuk menampilkan angka atau keluaran yang didapat dari proses logika fuzzy pada sistem. Maka dari itu dapat dikembangkan lebih lanjut untuk menggunakan media output yang sesungguhnya.

2.2 Dasar Teori

2.2.1 Logika Fuzzy

Logika *Fuzzy* pertama kali diperkenalkan oleh Lotfi A. Zadeh. *Fuzzy* berarti samar/tidak jelas. *Fuzzy logic* memiliki derajat keanggotaan dalam rentang 0 hingga 1, berbeda dengan logika digital/diskrit yang hanya memiliki dua nilai yaitu 0 atau 1. Logika *fuzzy* dapat mengolah nilai yang tidak pasti seperti “sangat”, “sedikit”, dan “kurang lebih”. Komputer tidak mengerti nilai asli dari kata-kata tersebut, sehingga dengan *Fuzzy logic*, komputer dapat mengolah ketidakpastian tersebut untuk memutuskan sesuatu yang membutuhkan kepintaran manusia dalam penalaran (Budiharto & Suhartono, 2014).

Pada teori himpunan klasik yang dapat disebut dengan himpunan *crisp* (himpunan tegas) hanya mengenal 2 kemungkinan pada fungsi keanggotaannya, yaitu kemungkinan yang termasuk dalam keanggotaan himpunan (logika 1) atau kemungkinan yang berada di luar keanggotaan (logika 0). Namun dalam teori himpunan *fuzzy* tidak hanya memiliki dua kemungkinan dalam menentukan sifat keanggotaannya tetapi memiliki sebuah derajat keanggotaan yang bernilai diantara 0 dan 1. Fungsi yang menetapkan nilai ini dinamakan fungsi keanggotaan yang disertakan dalam himpunan *fuzzy* (Purwaningrum, 2007).

2.2.1.1 Himpunan Fuzzy

Himpunan *fuzzy* merupakan pengelompokan sesuatu berdasarkan variabel bahasa (linguistik variabel), yang dinyatakan dengan fungsi keanggotaan, dalam semesta U . Keanggotaan suatu nilai pada himpunan dinyatakan dengan derajat keanggotaan yang nilainya antara 0.0 sampai 1.0.

Himpunan *fuzzy* didasarkan pada gagasan untuk memperluas jangkauan fungsi karakteristik sedemikian hingga fungsi tersebut akan mencakup bilangan real pada interval $[0, 1]$. Nilai keanggotaannya menunjukkan bahwa suatu *item* tidak hanya bernilai benar atau salah. Nilai 0 menunjukkan salah, nilai 1 menunjukkan benar, dan masih ada nilai-nilai yang terletak antara benar dan salah (Saelan, 2009).

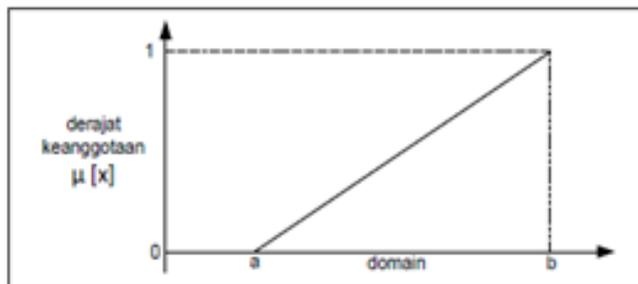
2.2.1.2 Fungsi Keanggotaan

Fungsi keanggotaan (membership function) adalah suatu kurva yang menunjukkan pemetaan titik-titik *input* data ke dalam nilai keanggotaan yang memiliki interval antara 0 sampai 1 (Susilo, 2006). Ada beberapa fungsi yang bisa digunakan diantaranya:

a. Representasi linear

Pada representasi linear, pemetaan *input* ke derajat keanggotaannya dapat digambarkan sebagai suatu garis lurus. Bentuk ini paling sederhana dan menjadi pilihan yang baik untuk mendekati suatu konsep yang kurang jelas. Ada 2 keadaan *fuzzy* linear yaitu representasi linear naik dan representasi linear turun. Representasi linear naik dimulai dari nilai *domain* yang memiliki nilai keanggotaan nol yang dilambangkan

dengan huruf “a” bergerak ke kanan menuju ke nilai *domain* yang memiliki derajat keanggotaan yang lebih tinggi yang dilambangkan huruf “b” dan untuk derajat keanggotaan pada kurva dilambangkan dengan $\mu[x]$ seperti pada Gambar 2.1 berikut:



Gambar 2.1 Representasi linear naik

Fungsi keanggotaan:

$$\mu[x] = \begin{cases} 0 & ; x \leq a \\ \frac{(x-a)}{(b-a)} & ; a \leq x \leq b \\ 1 & ; x \geq b \end{cases} \quad (2.1)$$

Representasi linear turun dimulai dari nilai *domain* dengan derajat keanggotaan tertinggi pada sisi kiri, kemudian bergerak turun ke nilai *domain* yang memiliki derajat keanggotaan lebih rendah seperti pada Gambar 2.2 berikut:



Gambar 2.2 Representasi linear turun

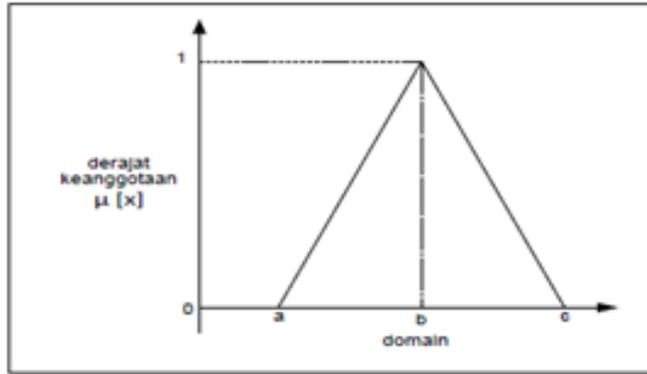
Fungsi keanggotaan:

$$\mu[x] = \begin{cases} \frac{(b-x)}{(b-a)} & ; a \leq x \leq b \\ 1 & ; x \geq b \end{cases} \quad (2.2)$$

b. Representasi Kurva Segitiga.

Representasi kurva segitiga merupakan gabungan dari dua representasi linear naik dan linear turun. Pada kurva segitiga memiliki 3 parameter yaitu a, b dan c yang digunakan untuk merepresentasikan fungsi keanggotaan himpunan normal seperti pada Gambar 2.3 berikut:





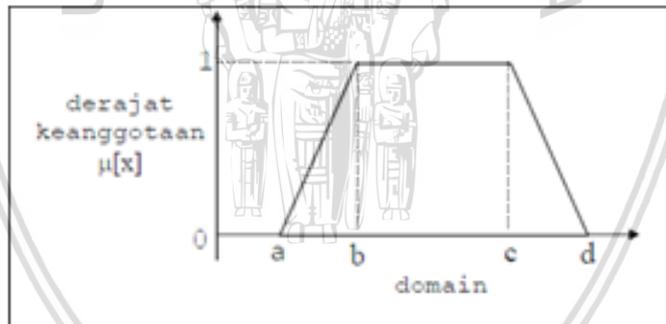
Gambar 2.3 Representasi kurva segitiga

Fungsi keanggotaan:

$$\mu[x] = \begin{cases} 0 & ; x \leq a \text{ atau } x \geq c \\ \frac{(x-a)}{(b-a)} & ; a \leq x \leq b \\ \frac{(c-x)}{(c-a)} & ; b \leq x \leq c \end{cases} \quad (2.3)$$

c. Representasi Kurva Trapesium.

Kurva trapesium memiliki 4 parameter yaitu a, b, c, dan d. Bentuk kurva ini sama seperti kurva segitiga, namun ada beberapa titik parameter yang memiliki nilai derajat keanggotaan 1 seperti pada Gambar 2.4 berikut:



Gambar 2.4 Representasi kurva trapesium

Fungsi keanggotaan:

$$\mu[x] = \begin{cases} 0 & ; x \leq a \text{ atau } x \geq d \\ \frac{(x-a)}{(b-a)} & ; a \leq x \leq b \\ 1 & ; b \leq x \leq c \\ \frac{(d-x)}{(d-c)} & ; c \leq x \leq d \end{cases} \quad (2.4)$$

2.2.1.3 Metode Mamdani

Metode Mamdani merupakan teknik atau metode *fuzzy* yang paling umum digunakan. Metode ini lebih sering dikenal dengan nama Metode Max-Min (Budiharto & Suhartono, 2014). Pada metode Mamdani, ada 3 tahap untuk mendapatkan *output*, yaitu:



1. Fuzzifikasi

Pada metode mamdani baik variabel input maupun output dibagi menjadi satu atau lebih himpunan fuzzy.

2. Inferensi

Inferensi merupakan proses pembentukan *rule* dan mencari *rule evaluation* dan *rule aggregation*. *Rule evaluation* merupakan pembentukan *rule* pada sisi *input* dan *rule aggregation* merupakan pembentukan *rule* untuk sisi *output*. Pada proses inferensi fungsi implikasi yang digunakan dalam pembentukan *rule* adalah Min. Selanjutnya komposisi aturan yang digunakan pada proses inferensi ada tiga metode yaitu, max, additive dan probabilistic OD (probor).

3. Defuzzifikasi

Input dari proses defuzzifikasi adalah suatu himpunan fuzzy yang diperoleh dari komposisi-komposisi aturan fuzzy, output yang dihasilkan merupakan suatu bilangan pada domain himpunan fuzzy tersebut. Sehingga jika diberikan suatu himpunan fuzzy dengan range tertentu, maka harus dapat diambil suatu nilai crisp/tegas tertentu sebagai nilai output. Ada beberapa metode defuzzifikasi yang dapat digunakan pada mamdani yaitu, *centroid*, *bisektor*, *mean of maximum*, *largest of maximum* dan *smallest of maximum*.

2.2.1.4 Embedded Fuzzy Logic Library (eFLL)

Embedded Fuzzy Logic Library (eFLL) yang dikembangkan oleh Robotic Research Group (RRG) di Universitas Negeri Piauí (UESPI-Teresina) merupakan *library* standar sistem *embedded* yang digunakan untuk mengimplementasikan sistem logika fuzzy secara mudah dan efisien. *Library* ini ditulis dalam bahasa C++/C, dan hanya menggunakan standar bahasa C dalam *library* "stdlib.h", sehingga eFLL di desain tidak hanya dapat digunakan pada arduino saja, melainkan semua sistem *embedded*. *Library* ini tidak memiliki batasan pada banyaknya *Fuzzy*, *Aturan Fuzzy*, *Input* atau , namun terbatas pada kekuatan proses dan kapasitas penyimpanan pada tiap mikrokontroler. eFLL menggunakan Metode Mamdani, dimana Max-Min sebagai inferensi dan *Center Of Area* untuk defuzzifikasinya (Alves & Lemos, 2016).

Berikut beberapa fungsi objek yang digunakan dalam *library* ini, diantaranya:

1. **Fuzzy object** – Objek ini mencakup semua sistem *Fuzzy*, di dalamnya dapat dimanipulasi sebuah *fuzzy set*, aturan linguistik, *input* dan .
2. **FuzzyInput object** – Objek ini digunakan untuk mengelompokkan semua *fuzzy set input* yang termasuk dalam *domain* yang sama.
3. **FuzzyOutput object** – Objek ini sama seperti *FuzzyInput object*, yang digunakan untuk mengelompokkan semua *fuzzy set output* yang termasuk dalam *domain* yang sama.

4. **FuzzySet object** – Objek ini merupakan objek utama dalam *fuzzy library*, yang mana tiap set mampu memodelkan sistem grafik *fuzzy*. Saat ini *library* mendukung fungsi keanggotaan segitiga, trapesium, dan *singleton*, yang dirancang berbasis *point* A, B, C, dan D, dan dilewatkan oleh parameter melalui konstruktor *FuzzySet(float A, float B, float C, float D)*.
5. **FuzzyRule object** – Objek ini digunakan untuk memasukkan basis aturan dari *fuzzy*.
6. **FuzzyRuleAntecedent object** – Objek ini digunakan untuk menggabungkan semua objek *rule* dari *input*, dan bertanggung jawab untuk merancang *antecedent* dari kondisi ekspresi *FuzzyRule*.
7. **FuzzyRuleConsequent object** – Objek ini digunakan sebagai konsekuensi atau output dari objek *rule*, dan bertanggung jawab untuk merancang ekspresi dari *FuzzyRule*.
8. **bool setInput(int id, float value);** – Fungsi ini digunakan untuk menghubungkan nilai *input Crisp* ke sistem, dimana parameter “id” merupakan *FuzzyInput object* yang merupakan nilai parameter yang dimaksud.
9. **bool fuzzify();** – Fungsi ini untuk menjalankan proses fuzzifikasi, komposisi aturan dan inferensi.
10. **float defuzzify(int id);** – Fungsi ini digunakan untuk proses akhir defuzzifikasi, dimana parameter “id” merupakan *Fuzzy object* yang akan didapatkan nilai defuzzifikasinya.
11. **float getPertinence();** – Fungsi ini digunakan untuk mencari tahu *fuzzy set* mana yang aktif.

2.2.2 Arduino Mega

Arduino Mega merupakan Mikrokontroler Board berbasis ATmega1280 yang memiliki 54 pin *input/output* digital. Selain itu Arduino Mega di desain untuk projek-projek yang kompleks dan juga memiliki kapasitas memori flash sebesar 128 KB yang mampu menampung banyak baris kode program. Dan mikrokontroler ini diprogram melalui aplikasi Arduino Software (IDE), aplikasi ini bersifat *free* yang dapat digunakan pada banyak operating sistem seperti Windows, Linux, dan MAC OS. Aplikasi Arduino IDE nantinya akan memudahkan proses implementasi sistem dan menanamkan logika *fuzzy* pada perangkat. Berikut Tabel 2.1 akan menunjukkan spesifikasi Arduino Mega 1280 dan Gambar 2.5 akan menunjukkan bentuk fisiknya.

Tabel 2.1 Spesifikasi Arduino Mega

Spesifikasi	Keterangan
Mikrokontroler	ATmega1280
Tegangan Operasi	5V
Tegangan <i>Input</i> (rekomendasi)	7-12V
Tegangan <i>Input</i> (batas)	6-20V
Pin Digital I/O	54 (15 diantaranya menyediakan PWM)
Pin Analog <i>Input</i>	16
Arus DC per Pin I/O	40 mA
Arus DC Pin 3.3V	50 mA
<i>Memori Flash</i>	128 KB dimana 4 KB digunakan oleh bootloader
SRAM	8 KB
EEPROM	4 KB
Kecepatan <i>Clock</i>	16 MHz



Gambar 2.5 Bentuk Fisik Arduino ATmega1280

Sumber: (Arduino, n.d.)

2.2.3 Sensor Ultrasonik HC-SR04

Sensor Ultrasonik merupakan sensor yang dapat mengukur jarak antara 2 cm sampai 400 cm. Sensor ini berbasis gelombang ultrasonik yang akan dipancarkan dan kemudian pantulan dari gelombang akan diterima kembali oleh *receiver* ultrasonik. Selanjutnya jarak benda akan ditentukan dengan melakukan perhitungan selisih antara waktu pengiriman gelombang dan waktu gelombang diterima. Berikut Tabel 4.2 akan menunjukkan spesifikasi sensor Ultrasonik HC-SR04 dan Gambar 2.8 akan menunjukkan bentuk fisiknya.

Tabel 2.2 Spesifikasi sensor ultrasonik HC-SR04

Spesifikasi	Keterangan
<i>Operating Voltage</i>	5V DC
<i>Operating Frequency</i>	40Hz
Sinyal input <i>Trigger</i>	10 μ TTL <i>pulse</i>
Sudut pengukuran	15°
Jarak Pengukuran	2cm – 400cm



Gambar 2.6 Bentuk fisik sensor ultrasonik

Sumber: (sparkfun.com, 2017)

2.2.4 Sensor DHT11

Sensor DHT11 menggunakan sensor kelembaban kapasitif dan termistor untuk mengukur udara di sekitarnya, dan memberikan keluaran sinyal digital pada pin data. Kelebihan dari sensor ini antara lain memiliki kualitas yang sangat baik, selain itu konsumsi daya yang digunakan sangat rendah, dan memiliki respon yang cepat pada pembacaan data. Berikut Tabel 4.2 akan menunjukkan spesifikasi dari sensor DHT11 dan Gambar 2.9 yang akan menunjukkan bentuk fisiknya.

Tabel 2.3 Spesifikasi sensor DHT11

Spesifikasi	Keterangan
<i>Power Supply</i>	3,5 ~ 5,5V DC
Jarak pengukuran	Kelembaban relatif 20-90% Suhu 0-50 °C
Resolusi	16bit
<i>Package</i>	4 pin
Akurasi	Kelembaban relatif $\pm 5\%$ Suhu $\pm 2^{\circ}\text{C}$
Waktu respon	1 / e (63%)

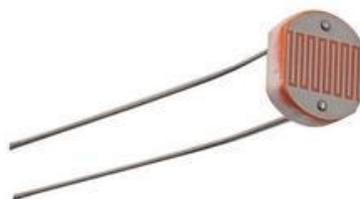


Gambar 2.7 Bentuk Fisik DHT11

Sumber: (tokopedia.com, 2017)

2.2.5 Sensor *Light Dependant Resistor* (LDR)

Light Dependant Resistor atau LDR digunakan untuk mengubah energi cahaya menjadi energi listrik. Resistansi/hambatan LDR berubah seiring dengan perubahan intensitas cahaya yang mengenainya. Dalam keadaan gelap, resistansi LDR sekitar $10\text{M}\Omega$ dan dalam keadaan terang sebesar $1\text{k}\Omega$ atau kurang. Waktu respon/komunikasi LDR adalah 20-30ms. Berikut Gambar 2.4 yang merupakan bentuk fisik dari sensor LDR.



Gambar 2.8 Bentuk fisik sensor LDR

Sumber: (tokopedia.com, 2017)

2.2.6 Driver Motor L298N

L298 adalah sirkuit monolitik terpadu dalam 15 timbal Multiwatt dan paket PowerSO20. Yang merupakan driver dual full-bridge tegangan tinggi yang dirancang untuk menerima tingkat standar logika TTL dan mendorong beban induktif seperti *relay*, solenoida, DC motor, dan motor stepper. Dua *input* masukan disediakan untuk dapat mengaktifkan dan menonaktifkan perangkat secara terpisah dari sinyal *input*. Tegangan minimal untuk masukan power antara 5V-35V, arus untuk masukan antara 0-36mA, dan memiliki daya maksimal yaitu 25W. Pin pada driver ini terdiri dari control pin yang berisi ENA (*Enable A*) digunakan untuk mengaktifkan data ke motor A dan ENB (*Enable B*) yang digunakan untuk mengaktifkan motor B (STMicroelectronics, n.d.). Berikut Gambar 2.9 akan menunjukkan bentuk fisik dari *driver motor* L298N.

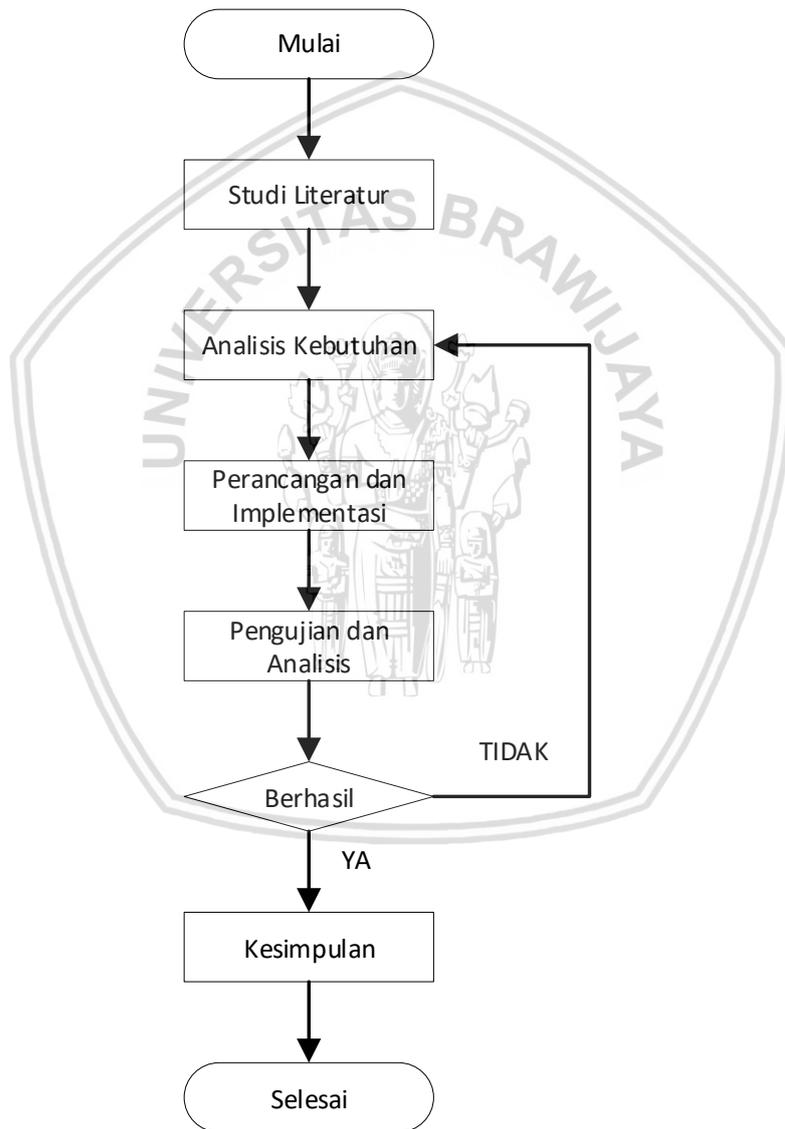


Gambar 2.9 Bentuk fisik *driver motor* L298N

Sumber: (dx.com, 2017)

BAB 3 METODOLOGI

Penelitian akan dimulai dengan studi literatur yang terkait dengan landasan pustaka dan dasar teori. Penelitian ini merupakan implementasi logika *fuzzy* untuk pengkondisian AC dan lampu. Penelitian bersifat implementatif karena hasil dari penelitian ini berupa produk perangkat prototype. Diawali dengan menentukan alur metode penelitian yang akan digunakan sebagai tahapan-tahapan yang akan ditempuh untuk menyelesaikan penelitian. Berikut ini merupakan diagram alir mengenai gambaran umum dalam pembuatan sistem yang digambarkan pada Gambar 3.1.



Gambar 3.1 Diagram Alir penelitian

3.1 Studi Literatur

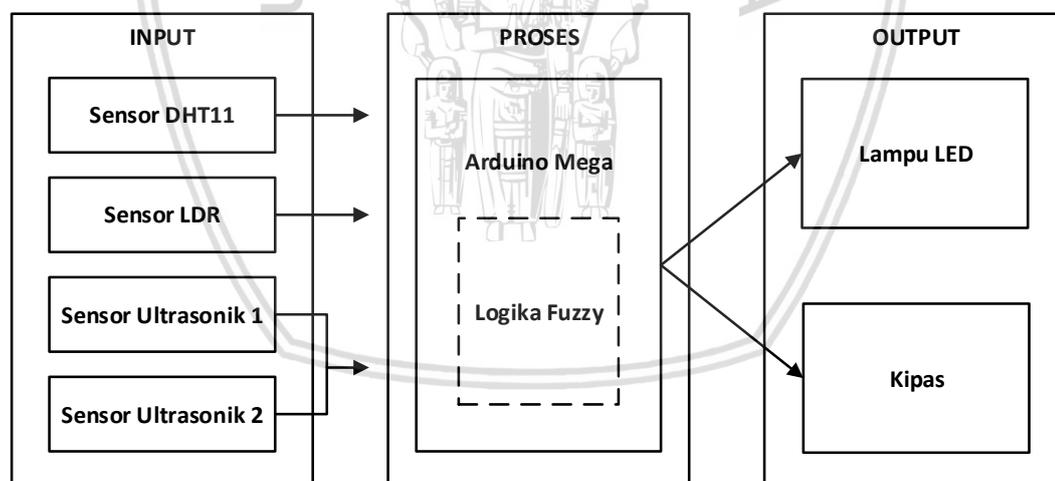
Studi literatur merupakan tahap melakukan pencarian referensi dan penyusunan dasar teori yang akan mendukung penelitian yang dibuat. Pada penelitian ini akan dijelaskan teori-teori dasar mengenai logika *fuzzy*, Arduino, Sensor LDR, sensor DHT, sensor Ultrasonik dan driver motor L298N.

3.2 Analisis Kebutuhan

Rekayasa kebutuhan adalah tahap yang bertujuan untuk melakukan analisis untuk kebutuhan yang diperlukan untuk melakukan penelitian. Adapun beberapa hal yang akan dibahas pada tahap ini antara lain adalah gambaran umum sistem yang akan menggambarkan secara umum keseluruhan sistem, analisis kebutuhan dilakukan untuk mengetahui apa saja yang diperlukan untuk kebutuhan penelitian seperti kebutuhan perangkat keras, kebutuhan perangkat lunak, kebutuhan fungsional dan kebutuhan non-fungsional pada sistem.

3.3 Perancangan dan Implementasi

Perancangan merupakan tahap yang akan dilakukan setelah analisis kebutuhan sistem. Tahap ini bertujuan untuk membuat gambaran mengenai implementasi metode logika fuzzy, implementasi perangkat keras dan perangkat lunak. Berikut Gambar 3.2 yang akan memperlihatkan diagram perancangan sistem.



Gambar 3.2 Diagram perancangan *input*, *proses*, dan *output*

Berdasarkan Gambar 3.2 diagram perancangan *input*, *proses* dan *ouput* pada sistem akan dijelaskan sebagai berikut.

1. *Input*, merupakan sebuah data masukan yang akan diproses oleh sistem, *input* yang digunakan terdiri dari empat sensor yaitu LDR, sensor DHT, dan dua sensor Ultrasonik.
2. *Proses*, merupakan pengolahan data masukan atau *input* yang nantinya diproses menjadi *output* . *Proses* pada sistem akan menggunakan logika *Fuzzy* yang akan dijalankan pada Arduino Mega.

3. *Output*, merupakan hasil dari proses dengan menggunakan media LED sebagai Lampu dan DC sebagai Kipas.

Selanjutnya setelah dilakukan perancangan akan dilakukan implementasi dari perancangan yang telah dibuat pada perangkat keras mulai dari implementasi logika fuzzy pada arduino Mega, implementasi sensor input dan sensor output.

3.4 Pengujian dan Analisis

Pada tahap pengujian sistem, dilakukan pengujian dengan beberapa skenario yaitu sebagai berikut.

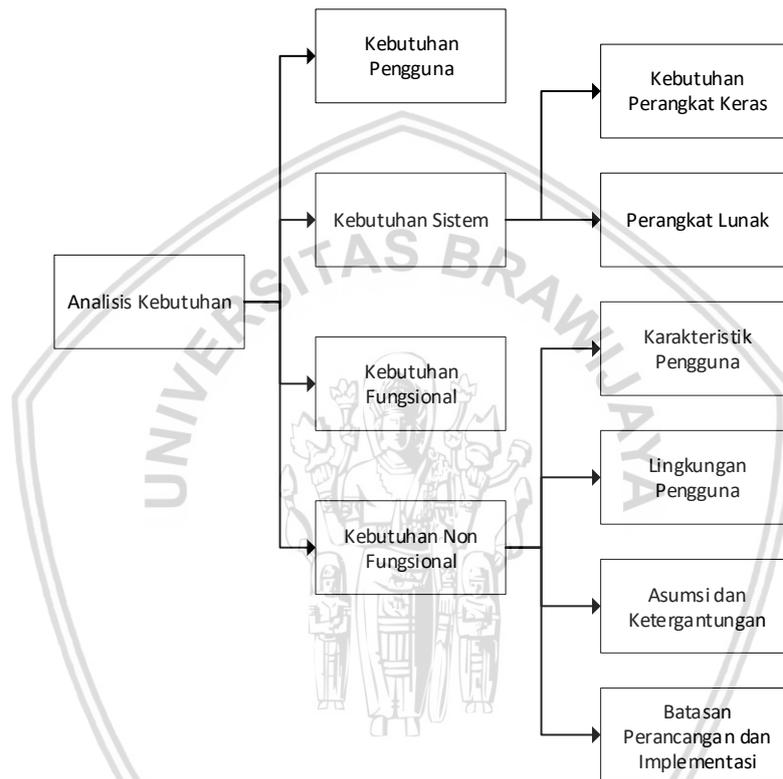
1. Pengujian akurasi keempat sensor; LDR, DHT11, dan Ultrasonik.
2. Pengujian kesesuaian *output* sistem dengan rule *Fuzzy* yang dibuat.
3. Pengujian akurasi kesesuaian *output* sistem dengan membandingkan nilai pada *output* MATLAB.

3.5 Kesimpulan dan Saran

Tahap pengambilan kesimpulan dilakukan setelah perancangan, implementasi, pengujian dan analisis telah dilakukan. Kesimpulan akan diambil berdasarkan hasil dari pengujian dan analisa sistem yang dibangun. Isi dari kesimpulan diharapkan dapat menjadi acuan untuk pengembangan dan penyempurnaan pada sistem. Pada tahap akhir penulisan adalah saran yang bertujuan untuk memperbaiki kesalahan dan kekurangan pada sistem yang telah dibuat.

BAB 4 ANALISIS KEBUTUHAN

Bab ini akan membahas beberapa kebutuhan yang diperlukan untuk melakukan penelitian ini, antara lain kebutuhan pengguna, kebutuhan sistem, kebutuhan fungsional, dan kebutuhan non fungsional. Ada 2 bagian pada kebutuhan sistem diantaranya kebutuhan perangkat keras dan perangkat lunak, lalu pada bagian kebutuhan non fungsional akan dibagi menjadi 4 diantaranya adalah karakteristik pengguna, lingkungan pengguna, asumsi dan ketergantungan, dan batasan perancangan dan implementasi.



Gambar 4.1 Diagram Analisis Kebutuhan

4.1 Kebutuhan Pengguna

Pada bagian ini akan dijelaskan mengenai gambaran bagaimana pengguna akan berinteraksi pada sistem yang dibuat agar sistem dapat berjalan sesuai dengan yang diinginkan. Pengguna akan berinteraksi dengan sistem melalui perangkat sensor yang akan mendeteksi pergerakan pengguna, selanjutnya perangkat akan memberikan hasil yang akan menyesuaikan dengan kondisi pergerakan pengguna, suhu, dan cahaya di sekitar perangkat sensor. Hasil akan dikeluarkan melalui prototipe kipas yang menggunakan motor dc dan prototipe lampu yang menggunakan LED.



4.2 Kebutuhan Sistem

Pada bab ini akan membahas kebutuhan yang diperlukan untuk membangun sistem, diantaranya adalah kebutuhan perangkat keras dan kebutuhan perangkat lunak. Untuk diagram blok lebih lengkap dari Gambar 3.2.

4.2.1 Kebutuhan Perangkat Keras

Perangkat keras yang akan digunakan dalam proses implementasi pada penelitian ini adalah sebagai berikut:

1. PC-Desktop

PC-Desktop akan digunakan untuk menuliskan kode program pada mikrokontroler Arduino Mega dengan aplikasi Arduino IDE. Arduino Mega akan dihubungkan melalui USB port pada PC-Desktop. Sistem operasi yang digunakan adalah Windows 10.

2. Arduino Mega

Pada penelitian ini mikrokontroler Arduino Mega digunakan sebagai pengembangan dari penelitian sebelumnya dari Mulia (2017) yang menggunakan mikrokontroler Arduino Uno yang memiliki kapasitas memori yang terbatas untuk penulisan baris kode program yang panjang. Kapasitas memori yang besar dari mikrokontroler Arduino Mega akan membantu dalam pembuatan kode program yang membutuhkan baris kode yang panjang, karena pada penelitian sistem akan menggunakan 135 *rule Fuzzy*. Mikrokontroler ini memiliki 54 pin dimana 14 pin-pinnya dapat digunakan sebagai *output* PWM dan 14 pin ini nantinya akan dimanfaatkan untuk *output* kipas dan lampu pada penelitian. Selain itu mikrokontroler ini diprogram melalui aplikasi Arduino Software (IDE), aplikasi ini bersifat *free* yang dapat digunakan pada banyak operating sistem seperti Windows, Linux, dan MAC OS. Aplikasi Arduino IDE nantinya akan memudahkan proses implementasi sistem dan menanamkan logika *fuzzy* pada perangkat pada penelitian ini. Untuk penjelasan spesifikasi Arduino Mega mengacu pada subbab 2.2.2.

3. Sensor Ultrasonik HC-SR04

Pada penelitian dibutuhkan 2 sensor ultrasonik HC-SR04 untuk mendeteksi gerak dan jarak seseorang pada ruangan. Digunakan sensor ultrasonik HC-SR04 sebagai pengembangan dari penelitian sebelumnya dari Mulia (2017) yang menggunakan sensor PIR yang hanya mengambil 2-bit data saja yaitu kondisi "0" dan "1". Pada penelitian ini kedua sensor bertujuan untuk mengambil nilai input data yang akan digunakan untuk sistem *fuzzy*. Kedua sensor ini akan diletakkan di bawah prototipe kipas untuk dapat mendeteksi pergerakan seseorang yang berada di depan prototipe kipas. Luas ruangan pada penelitian yang dilakukan adalah 3x3 meter. Sehingga sensor ini cocok untuk mendeteksi pergerakan seseorang

pada ruangan tersebut. Untuk penjelasan spesifikasi sensor ultrasonik mengacu pada subbab 2.2.3.

4. Sensor DHT11

Pada penelitian ini menggunakan sensor DHT11 untuk mendeteksi suhu pada ruangan. Alasan digunakan sensor DHT11 adalah jarak transmisi sinyal yang panjang mampu mendeteksi suhu pada area jangkauan di sekitar ruangan pengujian. Selain itu sensor DHT11 ini bersifat *ultra low-cost* karena pada mikrokontroler Arduino Mega menggunakan 5V sebagai tegangan operasinya dan DHT11 hanya membutuhkan tegangan 3V sampai 5V sebagai I/O. Nilai input suhu pada sensor nantinya akan digunakan pada sistem sebagai fungsi keanggotaan untuk menentukan hasil dan kondisi output kipas dan lampu. Untuk penjelasan spesifikasi sensor DHT11 mengacu pada subbab 2.2.4.

5. Sensor LDR (*Light Dependant Resistor*)

Pada penelitian ini akan digunakan sensor LDR untuk mendeteksi cahaya yang masuk pada ruangan. Nilai input pada sensor nantinya akan digunakan pada sistem sebagai fungsi keanggotaan untuk menentukan hasil dan kondisi output kipas dan lampu. Untuk penjelasan sensor LDR mengacu pada subbab 2.2.5.

6. LED

LED (*Light Emitted Diode*) adalah komponen elektronika yang mampu memancarkan sebuah cahaya monokromatik jika diberi tegangan *forward bias* (maju). Bentuk LED mirip dengan sebuah bohlam yang kecil dan mudah dipakai ke berbagai perangkat elektronika. Pada penelitian ini LED digunakan sebagai prototipe lampu yang akan menghasilkan keluaran kecerahan cahaya sesuai dengan hasil *output* pada sistem. Nilai keluaran kecerahan cahaya pada *output* diukur dengan menggunakan PWM (*Pulse Width Modulation*) pada program Arduino.

7. DC Motor

DC motor merupakan sebuah motor listrik yang bekerja menggunakan tegangan DC. DC motor ini memiliki panjang ukuran sekitar 29,5mm, poros melingkar yang kecil sekitar 2.8mm, dan tegangan operasi dari 1,5VDC sampai 4,5VDC. Pada penelitian ini DC motor akan digunakan sebagai prototipe kipas yang akan menghasilkan keluaran kecepatan putaran kipas sesuai dengan hasil *output* pada sistem. Nilai keluaran kecepatan putaran kipas pada *output* diukur dengan menggunakan PWM pada program Arduino.

8. Resistor

Resistor adalah sebuah komponen elektronika yang digunakan sebagai penghambat atau pembatas aliran listrik yang mengalir di suatu rangkaian elektronika, komponen ini bersifat resistif dan termasuk salah satu

komponen elektronika yang masuk dalam kategori komponen pasif. Kapasitas daya pada resistor berkisar 1/6 Watt – 5 Watt. Pada penelitian ini digunakan 2 resistor, salah satunya digunakan untuk hambatan pada LED dengan tujuan untuk menjaga LED agar tidak rusak, dan yang lain untuk mengkonversi data pengukuran input cahaya pada LDR menjadi lux.

9. Driver Motor L298N

Driver Motor L298N sebagai pengontrol putaran DC motor agar dapat diatur dengan PWM pada Arduino. Digunakan driver motor ini karena driver motor ini sangat umum digunakan untuk mengontrol kecepatan dan arah putaran dari DC motor dan driver motor ini mampu menggerakkan DC motor pada tegangan 5V sampai 35V. Untuk penjelasan spesifikasi mengacu pada subbab 2.2.6.

10. Baterai

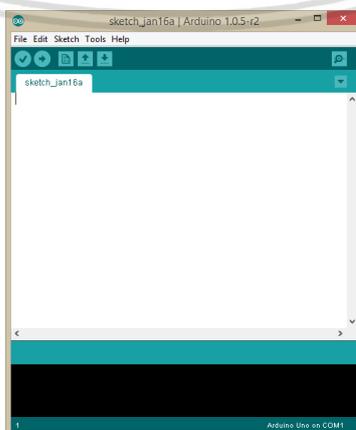
Baterai merupakan perangkat yang mengubah energi kimi menjadi energi listrik. Pada penelitian ini baterai digunakan bertegangan 9V untuk menggerakkan regulator pada driver motor L298N untuk dapat menggerakkan dc motor secara maksimal.

4.2.2 Kebutuhan Perangkat Lunak

Perangkat lunak yang akan digunakan dalam proses implementasi pada penelitian ini adalah sebagai berikut:

1. Arduino IDE

Arduino IDE merupakan perangkat lunak *open-source* Arduino yang digunakan untuk menuliskan kode program yang nantinya akan di tanam atau di *upload* pada mikrokontroler Arduino Mega. Aplikasi ini dapat berjalan di berbagai sistem operasi seperti Linux, Windows, dan MAC OS X. Aplikasi ini memudahkan penelitian untuk mengimplementasikan sistem logika Fuzzy pada mikrokontroler Arduino Mega. Berikut gambar 4.2 akan menunjukkan tampilan dari aplikasi Arduino IDE.



Gambar 4.2 Tampilan dari aplikasi Arduino IDE

2. *Embedded Fuzzy Logic Library* (eFLL)

EFLD digunakan pada penelitian untuk memudahkan implementasi logika *Fuzzy* pada aplikasi Arduino IDE. EFLD menggunakan metode *Fuzzy Mamdani* pada perhitungannya. Untuk penjelasan tentang library mengacu pada subbab 2.2.1.4.

4.3 Kebutuhan Fungsional

Kebutuhan fungsional merupakan kebutuhan dimana sistem dapat memberikan layanan yang sesuai dengan dibentuknya sistem tersebut kepada pengguna. Kebutuhan fungsional yang dibutuhkan, antara lain:

1. Perangkat sistem mampu mendeteksi pergerakan dan jarak seseorang dalam jangkauan sensornya. Pada penelitian ada 3 kondisi yang digunakan pada gerak yaitu tidak ada, pasif, dan aktif. Dan 3 kondisi pada jarak yaitu dekat, sedang, dan jauh
2. Perangkat sistem mampu mendeteksi suhu di dalam ruangan. Pada penelitian ini digunakan 3 kondisi pada suhu yaitu dingin, normal, dan panas.
3. Perangkat sistem mendeteksi cahaya yang masuk pada ruangan. Pada penelitian ini digunakan 5 kondisi pada cahaya yaitu sangat gelap, gelap, remang, terang, dan sangat terang.
4. Perangkat sistem yang sudah ditanamkan logika *Fuzzy* mampu memberikan hasil *output* yang mengkondisikan kecepatan kipas dan kecerahan cahaya lampu. Pada penelitian ada 3 kondisi yang digunakan pada kipas yaitu pelan, sedang, dan cepat. Dan 5 kondisi untuk lampu yaitu padam, sangat redup, redup, terang, dan sangat terang.

4.4 Kebutuhan Non Fungsional

4.4.1 Karakteristik Pengguna

Implementasi sistem menggunakan perangkat yang akan diletakkan pada sebuah ruangan dimana pengguna akan melakukan sebuah pergerakan baik itu secara pasif (diam, duduk, tidur, dll.) maupun aktif (bergerak, berjalan, dll.). Pengguna juga membantu pengambilan data input gerak dan jarak yang nantinya akan digunakan untuk pengkondisian kecepatan kipas dan kecerahan cahaya lampu.

4.4.2 Lingkungan Pengguna

Lingkungan pengguna berada pada sebuah ruangan yang berukuran 3x3 meter. Lingkungan tidak lebih dari 4 meter dikarenakan kemampuan sensor Ultrasonik yang digunakan dalam mendeteksi pergerakan hanya memiliki jangkauan 2 cm hingga 4 meter saja. Selanjutnya perangkat diletakkan dekat dengan jendela pada ruangan agar dapat mendeteksi input cahaya masuk pada ruangan. Pengujian perangkat juga akan dilakukan pada ruangan tersebut.

4.4.3 Batasan Desain Sistem

Sistem yang diimplementasikan memiliki beberapa batasan untuk mencapai tujuan yang diharapkan, yaitu:

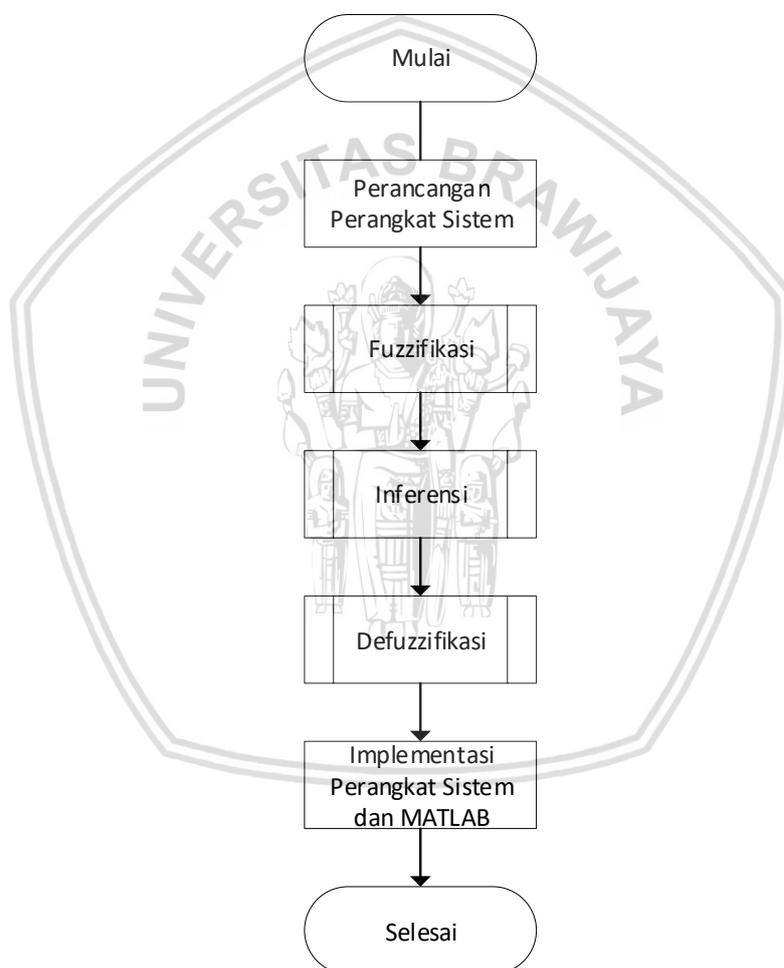
1. Implementasi perangkat sistem menggunakan prototipe di dalam sebuah kotak.
2. Logika *Fuzzy* diimplementasikan menggunakan *library* EFLL.
3. Keluaran output perangkat menggunakan prototipe LED sebagai lampu dan DC motor sebagai kipas.



BAB 5 PERANCANGAN DAN IMPLEMENTASI

5.1 Perancangan Sistem

Perangkat yang dirancang merupakan sistem yang dapat mengkondisikan kecepatan kipas dan kecerahan lampu berdasarkan *input* suhu, intensitas cahaya dan jarak atau pergerakan seseorang dalam ruangan. *Input* dari suhu, cahaya dan pergerakan akan diolah menggunakan mikrokontroller. Dalam mikrokontroller akan ditanam sebuah logika *fuzzy* yang akan memproses *input* untuk mengkondisikan *output* berupa kecepatan kipas dan kecerahan lampu. Berikut Gambar 5.1 akan menunjukkan diagram alir tahap-tahap yang akan dilakukan dalam membuat sistem.



Gambar 5.1 Diagram alir perancangan sistem

Seperti pada diagram alir 5.1 tahap pertama adalah perancangan perangkat sistem, pada tahap pertama akan dilakukan perancangan perangkat sensor *input*, sensor *output*, dan prototipe sistem. Tahap kedua adalah Fuzzifikasi, tahap ini akan dibuat perancangan fungsi keanggotaan dari tiap *input* mulai dari input gerak, jarak, suhu, dan cahaya kemudian dilakukan implementasi fungsi

keanggotaan *input* pada Arduino. Tahap ketiga adalah Inferensi, pada tahap ini akan dilakukan pembentukan rule *Fuzzy* sistem dan mengimplementasikannya pada Arduino. Tahap keempat adalah Defuzzifikasi, pada tahap ini akan dilakukan perancangan fungsi keanggotaan output dan implementasi defuzzifikasi pada Arduino. Tahap terakhir adalah Implementasi perangkat sistem dan MATLAB, pada tahap ini akan dilakukan implementasi perangkat sistem mulai dari perangkat *input* sensor, *output* kipas dan lampu, implementasi alat prototipe sistem dan implementasi MATLAB untuk pengujian sistem.

5.2 Perancangan Perangkat Sistem

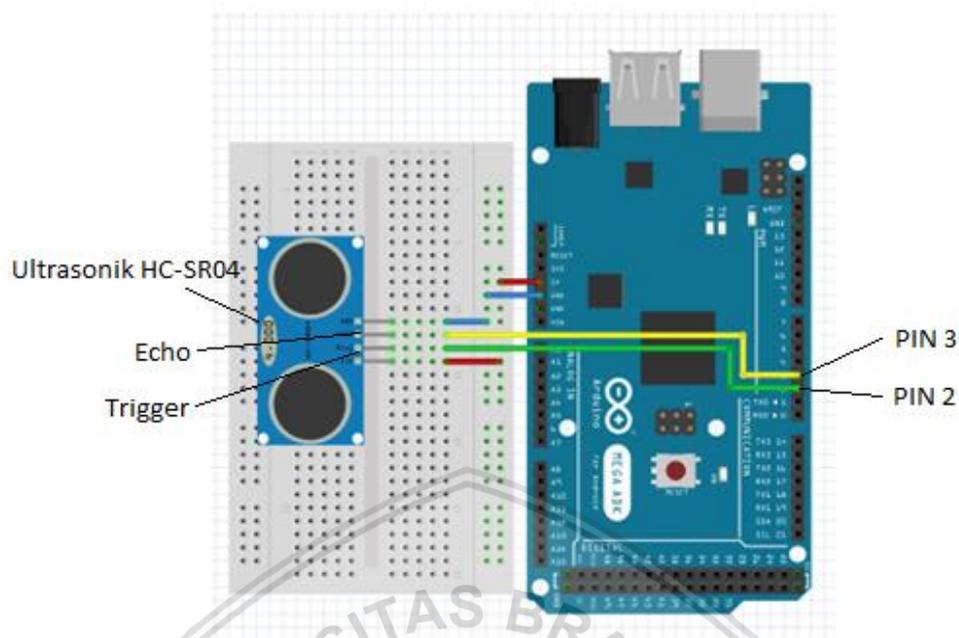
Pada bab ini akan membahas perancangan perangkat sistem mulai dari perancangan input sensor gerak, sensor jarak, sensor suhu, sensor cahaya, perancangan *output* kipas, *output* lampu, dan perancangan alat prototipe sistem.

5.2.1 Perancangan Input Sensor Gerak

Sensor gerak yang digunakan yaitu Ultrasonik HC-SR04 memiliki fungsi sebagai pendeteksi gerakan seseorang, diasumsikan tidak ada orang dalam ruangan jika sensor tidak mendeteksi apapun dalam jangkauannya, sensor mendeteksi gerakan pasif jika seseorang terdeteksi dalam jangkauan 0-200cm dan mendeteksi gerakan aktif jika seseorang terdeteksi dalam jangkauan 150-300cm. Berikut Tabel 5.1 akan menunjukkan hubungan pin sensor dengan arduino Mega dan Gambar 5.2 akan menunjukkan rangkaiannya. Sensor ultrasonik dihubungkan pada arduino melalui pin 2 untuk *trigger* dan pin 3 untuk *echo*, *trigger* digunakan untuk membangkitkan sinyal ultrasonik dan *echo* digunakan sebagai pendeteksi sinyal pantulan ultrasonik.

Tabel 5.1 Hubungan pin gerak dengan Arduino Mega

Pin Ultrasonik	Pin Arduino Mega
Vcc	5V
<i>Trigger</i>	2
<i>Echo</i>	3
GND	GND



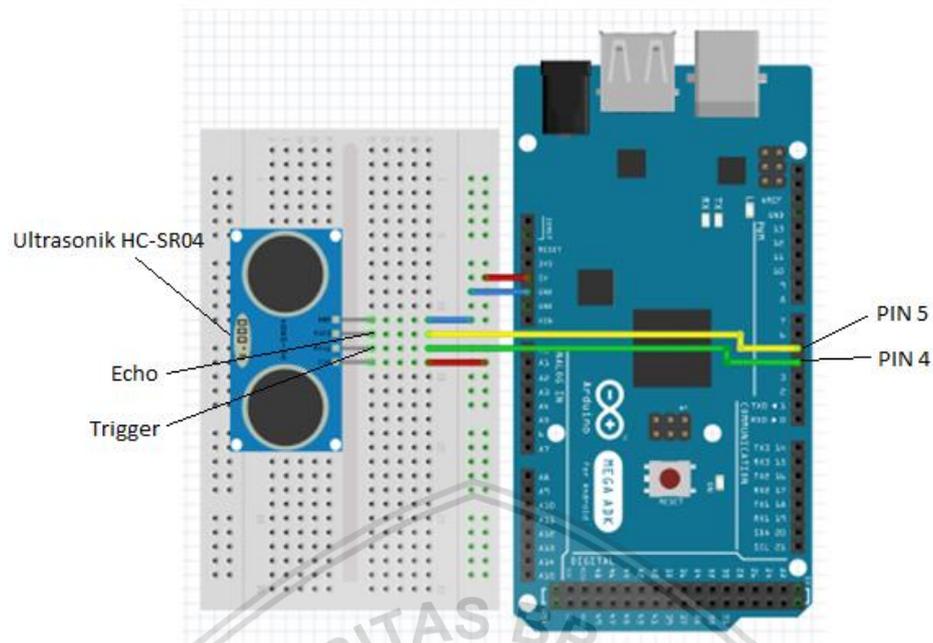
Gambar 5.2 Rangkaian Sensor Gerak

5.2.2 Perancangan Input Sensor Jarak

Sensor jarak yang digunakan sama seperti sensor gerak yaitu Ultrasonik HC-SR04 namun memiliki fungsi yang berbeda, sensor jarak digunakan untuk mengukur jarak seseorang dalam jangkauan sensor hingga 300 cm. Berikut Tabel 5.3 akan menunjukkan hubungan pin sensor dengan arduino Mega dan Gambar 5.2 akan menunjukkan rangkaiannya. Sensor ultrasonik dihubungkan pada arduino melalui pin 4 untuk *trigger* dan pin 5 untuk *echo*.

Tabel 5.2 Hubungan pin sensor jarak dengan Arduino Mega

Pin Ultrasonik	Pin Arduino Mega
Vcc	5V
<i>Trigger</i>	4
<i>Echo</i>	5
GND	GND



Gambar 5.3 Rangkaian Sensor Jarak

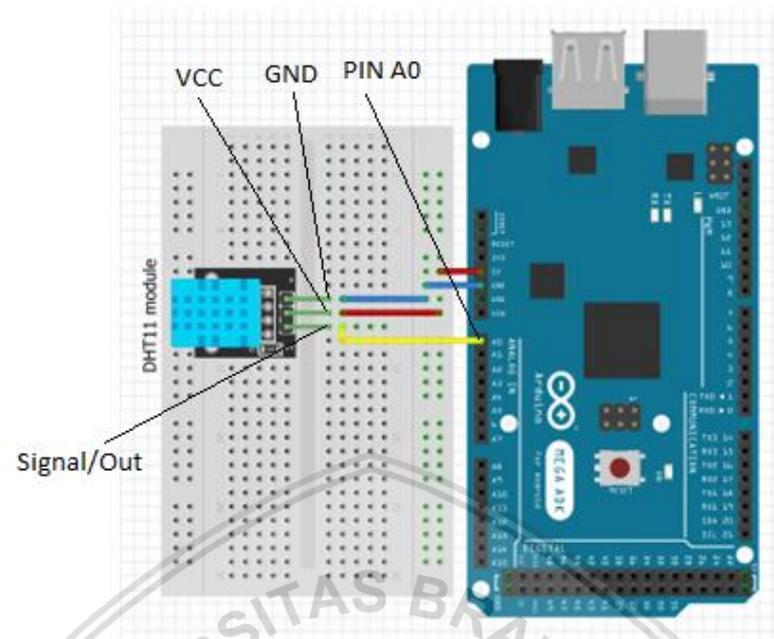
5.2.3 Perancangan Input Sensor Suhu

Sensor suhu yang digunakan adalah DHT11 module berfungsi untuk mengukur suhu di area sekitar jangkauan sensor. Sensor DHT11 merupakan jenis sensor digital yang mampu mendeteksi suhu dalam satuan °C. Pin Signal/Out pada DHT11 dihubungkan pada pin A0, pin ini merupakan pin analog yang dapat mengenali sinyal pada rentang nilai voltase yang digunakan, pin ini berguna untuk mengukur sensor-sensor input. Pin analog memiliki *converter* pada mikrokontroller yang dikenal dengan ADC (*Analog-to-Digital Converter*). *Converter* ini memiliki resolusi 10 bit yang akan menghasilkan nilai konversi dari 0 sampai 1023. Berikut Tabel 5.3 akan menunjukkan hubungan pin sensor dengan arduino Mega dan Gambar 5.4 akan menunjukkan rangkaiannya.

Tabel 5.3 Hubungan pin sensor suhu dengan Arduino Mega

Pin DHT11	Pin Arduino Mega
Vcc	5V
Signal/Out	A0
GND	GND





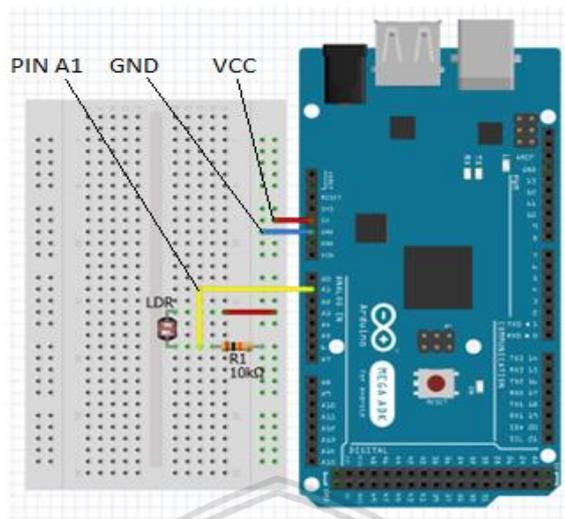
Gambar 5.4 Rangkaian Sensor Suhu

5.2.4 Perancangan Input Sensor Cahaya

Sensor cahaya yang digunakan adalah LDR yang berfungsi untuk mengukur intensitas cahaya di area sekitar jangkauan sensor. Untuk dapat mengambil nilai *input* dari sensor, sensor LDR akan dihubungkan dengan kabel jumper ke Arduino Mega. Berikut Tabel 5.4 akan menunjukkan hubungan pin yang digunakan dan Gambar 5.5 akan menunjukkan rangkaianannya. Pin Data pada LDR dihubungkan ke pin A1 yang merupakan pin analog, pin ini dapat mengenali sinyal masukan pada rentang nilai voltase yang digunakan.

Tabel 5.4 Hubungan pin sensor cahaya dengan Arduino Mega

Pin LDR	Pin Arduino Mega
Vcc	5V
Data	A1
GND	GND



Gambar 5.5 Rangkaian Sensor Cahaya

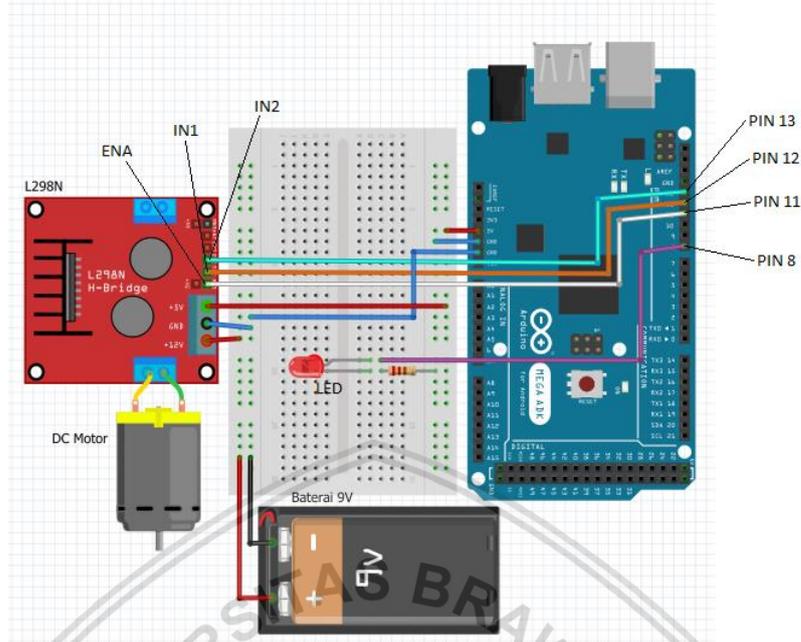
5.2.5 Perancangan Output Kipas dan Lampu

Alat yang digunakan sebagai prototipe output kipas adalah DC motor. Dengan *driver motor* L298N sebagai pengatur kecepatan kipas berdasarkan *input* dari arduino. Prototipe output lampu yang digunakan adalah LED. Berikut Tabel 5.5 yang merupakan hubungan pin *driver motor* L298N yang digunakan dan Gambar 5.6 akan menunjukkan rangkaian kipas dan lampu. Pin yang digunakan untuk menjalankan DC motor yaitu pin 11 untuk ENA, pin 12 untuk IN1, pin 13 untuk IN2, dan untuk menjalankan LED digunakan pin 8. Keempat pin ini merupakan pin PWM *output* 8-bit, dimana keluaran digital pada pin ini dapat dimanipulasi sedemikian rupa sehingga menghasilkan sinyal analog. Pin 12V pada driver motor L298N dihubungkan pada baterai 9V yang bertujuan untuk mengaktifkan regulator pada driver motor agar dapat menggerakkan DC motor secara maksimal.

Tabel 5.5 Hubungan pin *driver motor* dan pin LED dengan Arduino Mega

Pin L298N	Pin Arduino Mega
12Vdc	9V baterai
GND	GND & GND baterai
5Vdc	5V
ENA	11
IN1	12
IN2	13
IN3	-
IN4	-
ENB	-
LED	8

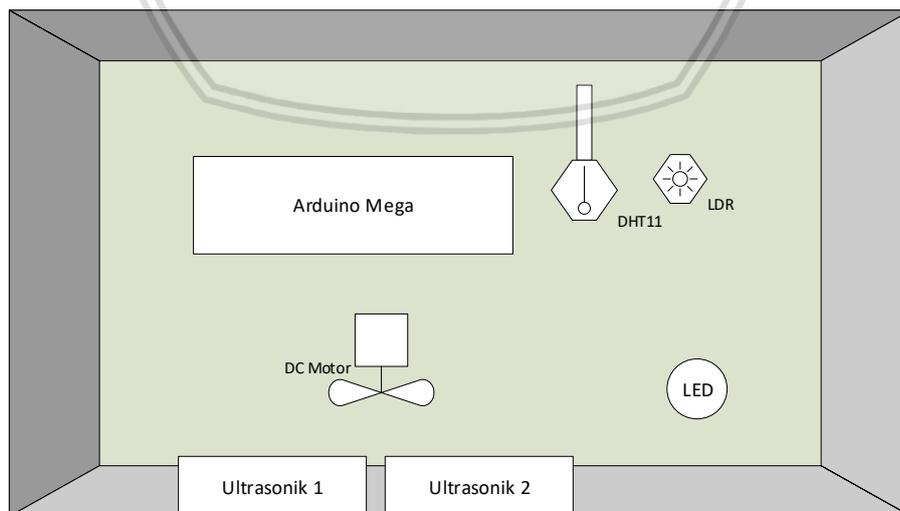




Gambar 5.6 Rangkaian *output* kipas dan lampu

5.2.6 Perancangan Alat Prototipe Sistem

Berikut perancangan alat prototipe yang akan dibuat dimana alat akan diletakkan pada sebuah kotak, DC motor akan digunakan sebagai kipas dan kemudian dua sensor ultrasonik akan diletakkan di bagian bawah kipas sebagai deteksi gerak dan jarak. Sensor DHT11 dan sensor LDR akan diletakkan di dalam kotak yang akan mendeteksi suhu dan cahaya yang masuk, sebuah LED juga diletakkan pada kotak yang akan digunakan sebagai lampu. Berikut Gambar 5.7 akan menunjukkan prototipe yang akan dibuat.

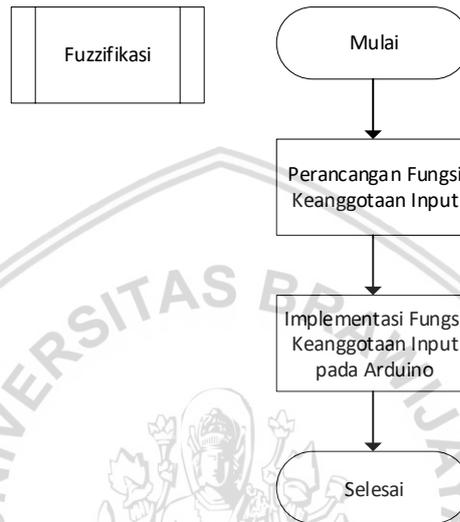


Gambar 5.7 Perancangan alat prototipe sistem



5.3 Fuzzifikasi

Pada bab ini akan ditentukan fungsi keanggotaan dari *input* melakukan pemetaan data tegas ke himpunan *fuzzy*. Tahap pertama pada fuzzifikasi adalah melakukan perancangan fungsi keanggotaan *input*, ada 4 fungsi yang digunakan diantaranya yaitu gerak, jarak, suhu dan cahaya. Kemudian setelah dibuat perancangannya fungsi keanggotaan *input* akan diimplementasikan pada Arduino dengan menggunakan fungsi dari *library* eFLL mengacu pada subbab 2.2.1.4. Berikut Gambar 5.8 akan menunjukkan diagram alir fuzzifikasi.



Gambar 5.8 Diagram alir proses fuzzifikasi

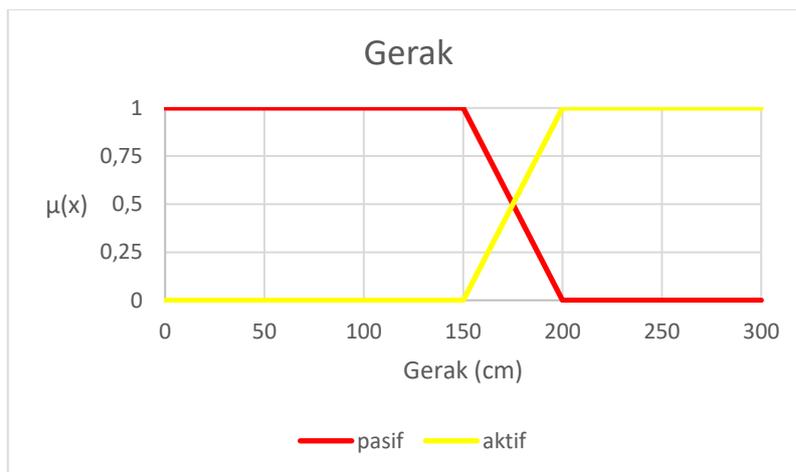
5.3.1 Perancangan Fungsi Keanggotaan Input

Pada bab ini akan dibahas perancangan fungsi keanggotaan *input* yang akan digunakan pada sistem, antara lain fungsi keanggotaan *input* gerak, jarak, suhu, dan cahaya.

5.3.1.1 Fungsi keanggotaan *input* gerak

Pada *input* gerak digunakan 3 fungsi keanggotaan yaitu tidak ada, pasif dan aktif. Digunakannya 3 fungsi keanggotaan sebagai pengembangan dari penelitian sebelumnya yang dilakukan oleh Mulia (2017) yang hanya menggunakan 2 fungsi keanggotaan pada pergerakan seseorang. Masing-masing memiliki *domain range* atau batasan nilai, diasumsikan tidak ada orang jika sensor tidak mendeteksi apapun dalam jangkauannya atau bernilai 0, dan pasif jika sensor mendeteksi seseorang dalam jarak 0-200cm, dan aktif jika sensor mendeteksi seseorang dalam jarak 200-300cm. Nilai data domain didapatkan melalui survey yang sudah dilakukan pada lampiran B. Fungsi keanggotaan *input* gerak dibentuk menggunakan representasi kurva trapesium karena *range* data yang diambil cukup banyak dan batasan pengambilan nilai input gerak mencapai 300cm. Ada 4 parameter yang digunakan pada kurva berdasarkan persamaan 2.4 pada subbab 2.2.1.2. Berikut Gambar 5.9 akan menunjukkan kurva dari fungsi keanggotaan *input* gerak.





Gambar 5.9 Kurva fungsi keanggotaan *input* gerak

Berikut Tabel 5.6 akan menunjukkan *domain range input* gerak.

Tabel 5.6 *Domain range input* gerak

No.	Fungsi Keanggotaan	Range (cm)
1	Tidak ada	0
2	Pasif	[0 200]
3	Aktif	[150 ≥200]

Berikut persamaan fungsi keanggotaan untuk *input* gerak.

$$\mu_{\text{Tidak ada}}[x] = 0 \tag{5.1}$$

$$\mu_{\text{Pasif}}[x] = \begin{cases} 1, & x \leq 150 \\ \frac{200-x}{200-150}, & 150 \leq x \leq 200 \\ 0, & x \geq 200 \end{cases} \tag{5.2}$$

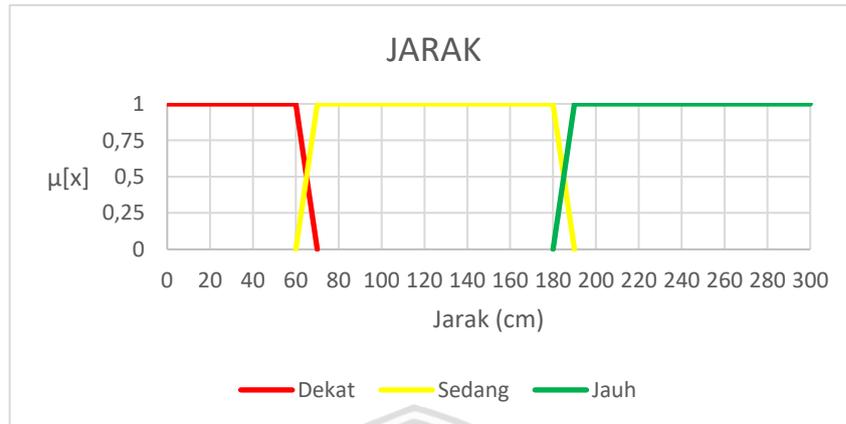
$$\mu_{\text{Aktif}}[x] = \begin{cases} 1, & x \geq 200 \\ \frac{x-150}{200-150}, & 150 \leq x \leq 200 \\ 0, & x \leq 150 \end{cases} \tag{5.3}$$

5.3.1.2 Fungsi keanggotaan *input* jarak

Pada *input* jarak terdapat 3 fungsi keanggotaan yaitu dekat, sedang dan jauh. Masing-masing memiliki *domain range* atau batasan nilai sesuai dengan data survey yang sudah dilakukan pada lampiran B. Fungsi keanggotaan *input* jarak dibentuk menggunakan representasi kurva trapesium karena *range* data yang diambil cukup banyak dan batasan pengambilan nilai input jarak mencapai 300cm. Ada 4 parameter yang diambil berdasarkan persamaan 2.4 pada subbab 2.2.1.2.



Berikut Gambar 5.10 akan menunjukkan kurva data *domain range* atau batasan nilai dari input jarak.



Gambar 5.10 Kurva fungsi keanggotaan *input* jarak

Berikut Tabel 5.7 akan menunjukkan *domain range input* jarak.

Tabel 5.7 *Domain range input* jarak

No.	Fungsi Keanggotaan	Range (cm)
1	Dekat	[0 ≤70]
2	Sedang	[60 190]
3	Jauh	[180 ≥190]

Berikut persamaan fungsi keanggotaan untuk *input* jarak.

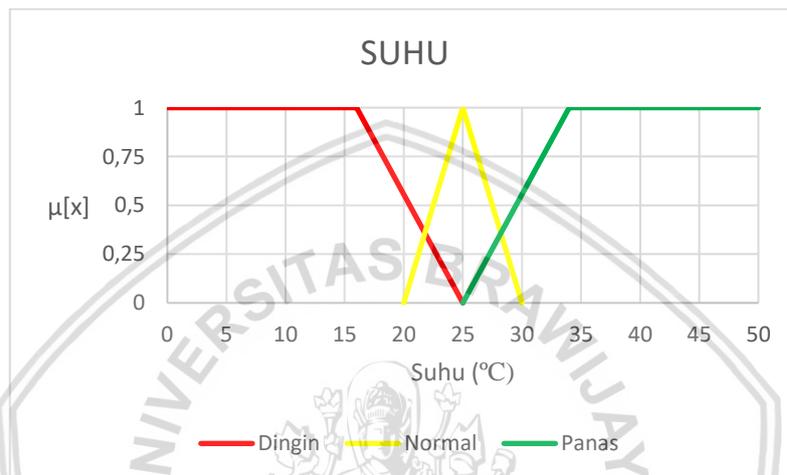
$$\mu_{\text{Dekat}}[x] = \begin{cases} 1, & x \leq 60 \\ \frac{70-x}{70-60}, & 60 \leq x \leq 70 \\ 0, & x \geq 70 \end{cases} \quad (5.4)$$

$$\mu_{\text{Sedang}}[x] = \begin{cases} 1, & x \geq 70 \text{ atau } x \leq 180 \\ \frac{x-60}{70-60}, & 60 \leq x \leq 70 \\ \frac{190-x}{190-180}, & 180 \leq x \leq 190 \\ 0, & x \leq 70 \text{ atau } x \geq 190 \end{cases} \quad (5.5)$$

$$\mu_{\text{Jauh}}[x] = \begin{cases} 1, & x \geq 190 \\ \frac{x-180}{190-180}, & 180 \leq x \leq 290 \\ 0, & x \leq 180 \end{cases} \quad (5.6)$$

5.3.1.3 Fungsi keanggotaan *input* suhu

Pada *input* suhu terdapat 3 fungsi keanggotaan yaitu dingin, normal dan jauh. Masing-masing memiliki *domain range* atau batasan nilai yang mengacu pada penelitian yang telah dilakukan oleh Mulia (2017) tentang pengkondisian AC dan lampu otomatis rada rebuah ruangan. Pada penelitian tersebut fungsi keanggotaan *input* suhu dibentuk menggunakan representasi kurva segitiga berdasarkan persamaan 2.3 dan kurva trapesium berdasarkan persamaan 2.4 pada subbab 2.2.1.2. Berikut Gambar 5.11 akan menunjukkan kurva *domain range* atau batasan nilai *input* suhu.



Gambar 5.11 Kurva fungsi keanggotaan *input* suhu

Berikut Tabel 5.8 akan menunjukkan *domain range input* suhu.

Tabel 5.8 Domain range *input* suhu

No.	Fungsi Keanggotaan	Range (°C)
1	Dingin	[≤16 25]
2	Normal	[20 30]
3	Panas	[25 ≥34]

Berikut persamaan fungsi keanggotaan untuk *input* suhu.

$$\mu_{\text{Dingin}}[x] = \begin{cases} 1, & x \leq 16 \\ \frac{25-x}{25-16}, & 16 \leq x \leq 25 \\ 0, & x \geq 25 \end{cases} \quad (5.7)$$

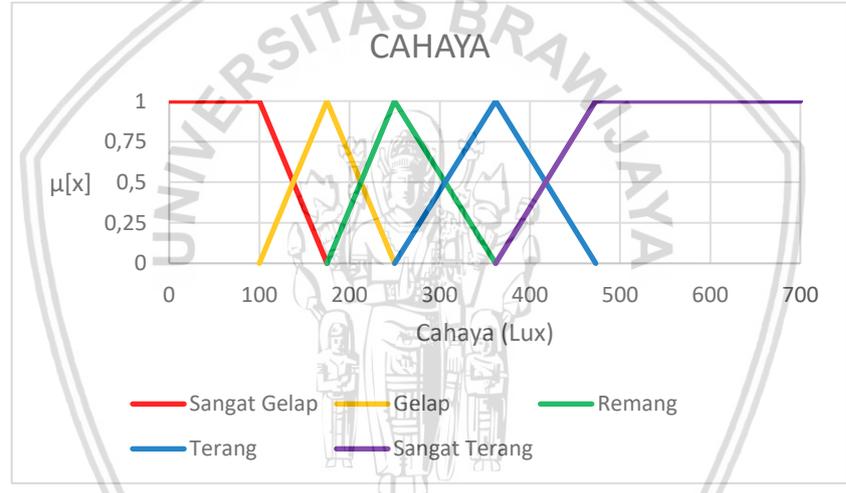
$$\mu_{\text{Normal}}[x] = \begin{cases} \frac{x-20}{25-20}, & 20 \leq x \leq 25 \\ \frac{30-x}{30-25}, & 25 \leq x \leq 30 \\ 0, & x \leq 20 \text{ atau } x \geq 30 \end{cases} \quad (5.8)$$



$$\mu_{Panas}[x] = \begin{cases} 1, & x \geq 34 \\ \frac{x-25}{34-25}, & 25 \leq x \leq 34 \\ 0, & x \leq 25 \end{cases} \quad (5.9)$$

5.3.1.4 Fungsi keanggotaan *input* cahaya

Pada *input* cahaya terdapat 5 fungsi keanggotaan yaitu sangat gelap, gelap, remang, terang dan sangat terang. Masing-masing memiliki *domain range* atau batasan nilai yang mengacu pada penelitian yang telah dilakukan oleh Caldo (2015) mengenai perancangan dan pengembangan logika *fuzzy* pada sistem pencahayaan menggunakan mikrokontroler Arduino. Pada penelitian tersebut fungsi keanggotaan *input* cahaya dibentuk menggunakan representasi kurva segitiga berdasarkan persamaan 2.3 dan kurva trapesium berdasarkan persamaan 2.4 pada subbab 2.2.1.2. Berikut Gambar 5.12 akan menunjukkan kurva *domain range* atau batasan nilai *input* cahaya.



Gambar 5.12 Kurva fungsi keanggotaan *input* cahaya

Berikut Tabel 5.9 akan menunjukkan *domain range input* cahaya.

Tabel 5.9 *Domain range input* cahaya

No.	Fungsi Keanggotaan	Range (lux)
1	Sangat Gelap	[0 ≤175]
2	Gelap	[100 250]
3	Remang	[175 362]
4	Terang	[250 473]
5	Sangat Terang	[362 ≥700]

Berikut persamaan fungsi keanggotaan untuk *input* cahaya.

$$\mu_{\text{Sangat Gelap}}[x] = \begin{cases} 1, & x \leq 100 \\ \frac{175-x}{175-100}, & 100 \leq x \leq 175 \\ 0, & x \geq 175 \end{cases} \quad (5.10)$$

$$\mu_{\text{Gelap}}[x] = \begin{cases} \frac{x-100}{175-100}, & 100 \leq x \leq 175 \\ \frac{250-x}{250-175}, & 175 \leq x \leq 250 \\ 0, & x \leq 100 \text{ atau } x \geq 250 \end{cases} \quad (5.11)$$

$$\mu_{\text{Remang}}[x] = \begin{cases} \frac{x-175}{250-175}, & 175 \leq x \leq 250 \\ \frac{362-x}{362-250}, & 250 \leq x \leq 362 \\ 0, & x \leq 175 \text{ atau } x \geq 362 \end{cases} \quad (5.12)$$

$$\mu_{\text{Terang}}[x] = \begin{cases} \frac{x-250}{362-250}, & 250 \leq x \leq 362 \\ \frac{473-x}{473-362}, & 362 \leq x \leq 473 \\ 0, & x \leq 250 \text{ atau } x \geq 473 \end{cases} \quad (5.13)$$

$$\mu_{\text{Sangat Terang}}[x] = \begin{cases} 1, & x \geq 473 \\ \frac{x-362}{473-362}, & 362 \leq x \leq 473 \\ 0, & x \leq 362 \end{cases} \quad (5.14)$$

5.3.2 Implementasi Fungsi Keanggotaan Input pada Arduino

Pada penelitian ini digunakan fungsi dari library eFLL untuk mempermudah implementasi fungsi keanggotaan *input* pada Arduino, untuk menentukan *domain range* atau batasan nilai akan digunakan fungsi *FuzzySet input* pada library eFLL mengacu pada subbab 2.2.1.4. Data *domain range* atau batasan nilai pada *FuzzySet input* mengacu pada subbab 5.3.1 mengenai perancangan fungsi keanggotaan input. Berikut Tabel 5.10 akan menunjukkan kode sumber dari fungsi *FuzzySet input*. Pada baris ke-1 sampai ke-3 merupakan kode sumber batasan nilai dari *input* gerak, baris ke-5 sampai ke-7 adalah kode sumber batasan nilai dari *input* jarak, baris ke-9 sampai baris ke-11 adalah kode sumber batasan nilai dari *input* suhu, dan baris ke-13 sampai baris ke-17 adalah kode sumber batasan nilai dari *input* cahaya.

Tabel 5.10 Kode sumber fungsi *FuzzySet input*

No	Kode Sumber
1	<code>FuzzySet* tdkada = new FuzzySet(0, 0, 0, 0);</code>
2	<code>FuzzySet* pasif = new FuzzySet(0, 0, 150, 200);</code>
3	<code>FuzzySet* aktif = new FuzzySet(150, 200, 300, 300);</code>
4	
5	<code>FuzzySet* dekat = new FuzzySet(0, 0, 60, 70);</code>
6	<code>FuzzySet* sedang = new FuzzySet(60, 70, 180, 190);</code>
7	<code>FuzzySet* jauh = new FuzzySet(180, 190, 300, 300);</code>
8	
9	<code>FuzzySet* dingin = new FuzzySet(0, 0, 16, 25);</code>
10	<code>FuzzySet* normal = new FuzzySet(20, 25, 25, 30);</code>
11	<code>FuzzySet* panas = new FuzzySet(25, 34, 40, 40);</code>
12	
13	<code>FuzzySet* sgelap = new FuzzySet(0, 0, 100, 175);</code>
14	<code>FuzzySet* gelap = new FuzzySet(100, 175, 175, 250);</code>
15	<code>FuzzySet* remang = new FuzzySet(175, 250, 250, 362);</code>
16	<code>FuzzySet* terang = new FuzzySet(250, 362, 362, 473);</code>
17	<code>FuzzySet* sterang = new FuzzySet(362, 473, 700, 700);</code>

Selanjutnya akan ditentukan variabel *fuzzy* dengan menggunakan fungsi *FuzzyInput object* mengacu pada subbab 2.2.1.4. Data variabel *fuzzy input* yang di dapatkan pada kode sumber mengacu pada subbab 5.3.1 mengenai perancangan fungsi keanggotaan *input*. Berikut Tabel 5.11 akan menunjukkan kode sumber dari fungsi *FuzzyInput object*. Pada baris ke-1 sampai ke-5 merupakan kode sumber fungsi keanggotaan dari *input* gerak, baris ke-7 sampai ke-10 adalah kode sumber fungsi keanggotaan dari *input* jarak, baris ke-13 sampai ke-17 adalah kode sumber fungsi keanggotaan dari *input* suhu, dan baris ke-19 sampai ke-25 adalah kode sumber fungsi keanggotaan dari *input* cahaya.

Tabel 5.11 Kode sumber fungsi *FuzzyInput object*

No	Kode Sumber
1	<code>FuzzyInput* gerak = new FuzzyInput(1);</code>
2	<code>gerak->addFuzzySet(tdkada);</code>
3	<code>gerak->addFuzzySet(pasif);</code>
4	<code>gerak->addFuzzySet(aktif);</code>
5	<code>fuzzy->addFuzzyInput(gerak);</code>
6	
7	<code>FuzzyInput* jarak = new FuzzyInput(2);</code>
8	<code>jarak->addFuzzySet(dekat);</code>
9	<code>jarak->addFuzzySet(sedang);</code>
10	<code>jarak->addFuzzySet(jauh);</code>
11	<code>fuzzy->addFuzzyInput(jarak);</code>
12	
13	<code>FuzzyInput* suhu = new FuzzyInput(3);</code>
14	<code>suhu->addFuzzySet(dingin);</code>
15	<code>suhu->addFuzzySet(normal);</code>
16	<code>suhu->addFuzzySet(panas);</code>
17	<code>fuzzy->addFuzzyInput(suhu);</code>
18	
19	<code>FuzzyInput* cahaya = new FuzzyInput(4);</code>
20	<code>cahaya->addFuzzySet(sgelap);</code>
21	<code>cahaya->addFuzzySet(gelap);</code>
22	<code>cahaya->addFuzzySet(remang);</code>
23	<code>cahaya->addFuzzySet(terang);</code>
24	<code>cahaya->addFuzzySet(sterang);</code>
25	<code>fuzzy->addFuzzyInput(cahaya);</code>



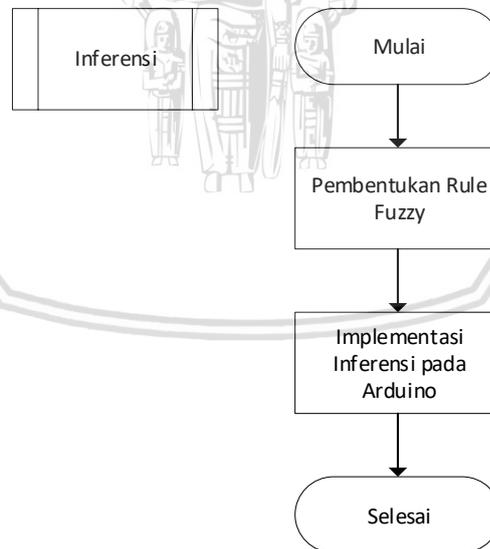
Terakhir akan mengeksekusi fuzzifikasi dengan menggunakan fungsi yaitu `bool fuzzify()` mengacu pada subbab 2.2.1.4. Berikut Tabel 5.12 akan menunjukkan kode sumber fungsi fuzzifikasi. Pada kode sumber baris ke-1 dan ke-2 *input* “ping1” dan “ping2” merupakan nilai data yang akan diambil dari pembacaan sensor ultrasonik, “ping1” akan mengambil nilai data gerak dan “ping2” akan mengambil nilai data jarak. Pada baris ke-3 merupakan kode sumber *input* dari sensor suhu, nilai data *input* “suhu” akan diambil dari pembacaan sensor DHT11, baris ke-4 adalah kode sumber *input* dari sensor cahaya yang akan diambil dari nilai data sensor LDR. Baris ke-5 adalah kode sumber untuk mengeksekusi proses fuzzifikasi.

Tabel 5.12 Kode sumber fungsi fuzzifikasi

No	Kode Sumber
1	<code>fuzzy->setInput(1, ping1);</code>
2	<code>fuzzy->setInput(2, ping2);</code>
3	<code>fuzzy->setInput(3, suhu);</code>
4	<code>fuzzy->setInput(4, lux);</code>
5	<code>fuzzy->fuzzify();</code>

5.4 Inferensi

Proses kedua yaitu inferensi, proses ini merupakan pembentukan *rule* dan mencari *rule evaluation* dan *rule aggregation*. Pada proses inferensi akan dilakukan pembentukan rule Fuzzy. Selanjutnya dilakukan implementasi inferensi pada Arduino. Berikut Gambar 5.13 akan menunjukkan diagram alir proses inferensi.



Gambar 5.13 Diagram Alir Inferensi

5.4.1 Pembentukan Rule Fuzzy

Pembentukan rule Fuzzy digunakan untuk menyatakan relasi antara variabel *input* dan *output*. Pada penelitian ini kondisi *output* yang digunakan ada 4 kondisi untuk kecepatan kipas yaitu padam, pelan, sedang, cepat, dan ada 5 kondisi kecerahan lampu yaitu padam, sangat redup. Jumlah *rule* didapat dari nilai

perkalian jumlah fungsi keanggotaan variabel *input*. Ada 3 jumlah fungsi keanggotaan pada variabel gerak, 3 pada variabel jarak, 3 pada variabel suhu, dan 4 pada variabel cahaya. Jadi, ada 135 *rule* yang akan dibuat sehingga membentuk beberapa kombinasi *rule fuzzy*. Berikut Tabel 5.13 akan menunjukkan potongan daftar *Rule Fuzzy input* beserta *output*-nya. Untuk daftar *rule fuzzy* yang lebih lengkap dapat dilihat pada lampiran A.

Tabel 5.13 Potongan Daftar Rule Fuzzy

No.	Input				Output	
	Gerak	Jarak	Suhu	Cahaya	Kipas	Lampu
1	Tidak Ada	Dekat	Dingin	Sangat Gelap	Padam	Padam
2	Tidak Ada	Dekat	Dingin	Gelap	Padam	Padam
3	Tidak Ada	Dekat	Dingin	Remang	Padam	Padam
4	Tidak Ada	Dekat	Dingin	Terang	Padam	Padam
5	Tidak Ada	Dekat	Dingin	Sangat Terang	Padam	Padam
...
...
...
133	Aktif	Jauh	Panas	Remang	Cepat	Redup
134	Aktif	Jauh	Panas	Terang	Cepat	Sangat Redup
135	Aktif	Jauh	Panas	Sangat Terang	Cepat	Padam

5.4.2 Implementasi Inferensi pada Arduino

Setelah dilakukan pembentukan *rule Fuzzy*, semua *rule* yang telah dibuat akan diimplementasikan pada Arduino dengan menggunakan fungsi *FuzzyRuleAntecedent object* mengacu pada subbab 2.2.1.4, fungsi ini digunakan untuk menentukan hubungan aturan dari tiap anggota variabel *input* gerak, jarak, suhu dan cahaya, *rule Fuzzy* dan hubungan aturan yang digunakan. Selanjutnya semua *antecedent* akan digabungkan menggunakan fungsi implikasi Min. Berikut Tabel 5.14 akan menunjukkan beberapa potongan kode sumber *FuzzyRuleAntecedent object*. Pada baris ke-1 sampai baris ke-6 adalah kode sumber *FuzzyRuleAntecedent* dari fungsi keanggotaan *input* gerak, pada baris ke-7 sampai ke-8 adalah kode sumber fungsi implikasi Min. Pada baris ke-11 sampai baris ke-16 merupakan penggabungan keseluruhan kode sumber *FuzzyRuleAntecedent* menjadi 1 *rule*.



Tabel 5.14 Potongan kode sumber fungsi *FuzzyRuleAntecedent* object

No	Kode Sumber
1	FuzzyRuleAntecedent* TidakAdaAndDekat = new
2	FuzzyRuleAntecedent();
3	FuzzyRuleAntecedent* TidakAdaAndSedang = new
4	FuzzyRuleAntecedent();
5	FuzzyRuleAntecedent* TidakAdaAndJauh = new
6	FuzzyRuleAntecedent();
7	TidakAdaAndDekat->joinWithAND(tdkada, dekat);
8	TidakAdaAndSedang->joinWithAND(tdkada, sedang);
9	TidakAdaAndJauh->joinWithAND(tdkada, jauh);
10	
11	FuzzyRuleAntecedent* Rule1 = new FuzzyRuleAntecedent();
12	FuzzyRuleAntecedent* Rule2 = new FuzzyRuleAntecedent();
13	FuzzyRuleAntecedent* Rule3 = new FuzzyRuleAntecedent();
14	Rule1->joinWithAND(TidakAdaAndDekat, DinginAndSangatGelap);
15	Rule2->joinWithAND(TidakAdaAndDekat, DinginAndGelap);
16	Rule3->joinWithAND(TidakAdaAndDekat, DinginAndRemang);

Selanjutnya menentukan *FuzzyRuleConsequent* object. Fungsi ini bertujuan untuk menentukan hubungan aturan dari anggota variabel *output* kipas dan lampu. Berikut Tabel 5.15 akan menunjukkan potongan kode sumber dari fungsi *FuzzyRuleConsequent*. Pada baris ke-1 sampai baris ke-10 adalah kode sumber *FuzzyRuleConsequent* dari fungsi keanggotaan *output* lampu.

Tabel 5.15 Potongan kode sumber fungsi *FuzzyRuleConsequent*

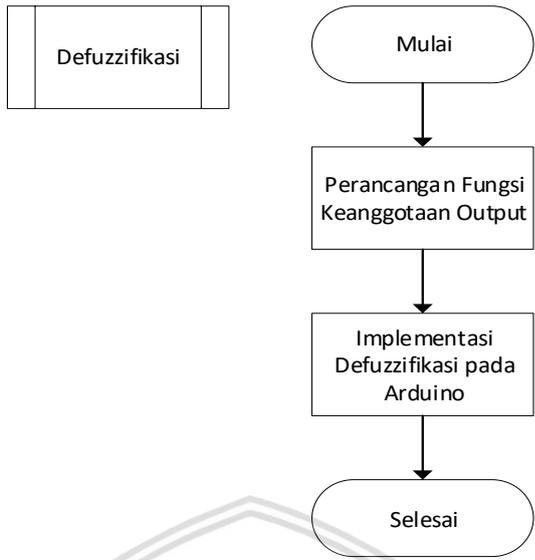
No	Kode Sumber
1	FuzzyRuleConsequent* PelanAndPadam = new
2	FuzzyRuleConsequent();
3	FuzzyRuleConsequent* PelanAndSangatRedup = new
4	FuzzyRuleConsequent();
5	FuzzyRuleConsequent* PelanAndRedup = new
6	FuzzyRuleConsequent();
7	FuzzyRuleConsequent* PelanAndTerang = new
8	FuzzyRuleConsequent();
9	FuzzyRuleConsequent* PelanAndSangatTerang = new
10	FuzzyRuleConsequent();

Dan yang terakhir yaitu mengeksekusi proses inferensi dengan fungsi dan kode sumber yang sama dengan proses eksekusi fuzzifikasi pada subbab 5.3.2 Tabel 5.12.

5.5 Defuzzifikasi

Proses selanjutnya yaitu defuzzifikasi, yaitu proses akhir yang akan mengkonversi semua nilai *input crisp* menjadi nilai tegas. Proses defuzzifikasi diawali dengan melakukan perancangan fungsi keanggotaan *output* dan selanjutnya defuzzifikasi akan diimplementasikan pada Arduino. Berikut Gambar 5.4 akan menunjukkan diagram alir proses defuzzifikasi.





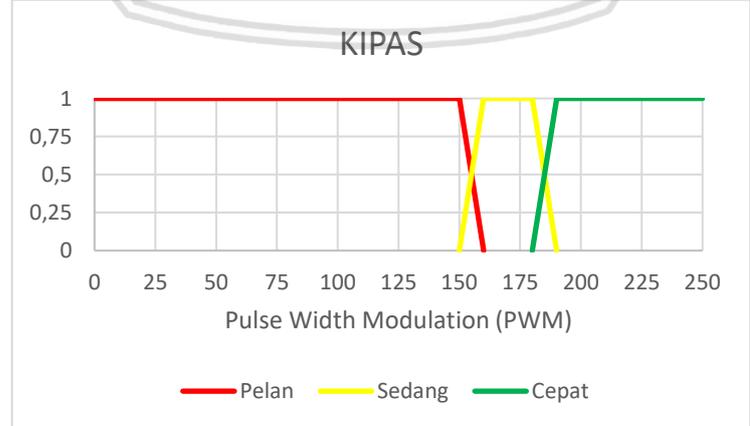
Gambar 5.14 Diagram alir defuzzifikasi

5.5.1 Perancangan Fungsi Keanggotaan Output

Pada bab ini akan dibahas mengenai perancangan fungsi keanggotaan *output* yang digunakan pada sistem, antara lain fungsi keanggotaan *output* kipas dan *output* lampu.

5.5.1.1 Fungsi keanggotaan *output* kipas

Pada *output* kipas terdapat 3 fungsi keanggotaan yaitu pelan, sedang, dan cepat. Masing-masing memiliki *domain range* atau batasan nilai yang mengacu berdasarkan data survey yang telah dilakukan oleh penulis pada lampiran C. Fungsi keanggotaan *output* kipas dibentuk menggunakan representasi kurva trapesium karena data PWM kecepatan dan kecerahan cukup banyak dengan batasan nilai PWM mencapai 250. Ada 4 parameter yang digunakan berdasarkan persamaan 2.4 pada subbab 2.2.1.2. Berikut Gambar 5.9 akan menunjukkan kurva *domain range* atau batasan nilai *output* kipas.



Gambar 5.15 Kurva fungsi keanggotaan *output* kipas



Berikut Tabel 5.5 akan menunjukkan *domain range* dari *output* kipas.

Tabel 5.16 Domain range output kipas

No.	Fungsi Keanggotaan	Range (PWM)
1	Pelan	[0 ≤160]
2	Sedang	[150 190]
3	Cepat	[180 ≥190]

Berikut persamaan fungsi keanggotaan untuk *output* kipas.

$$\mu_{\text{Pelan}}[x] = \begin{cases} 1, & x \leq 150 \\ \frac{160-x}{160-150}, & 150 \leq x \leq 160 \\ 0, & x \geq 160 \end{cases} \quad (5.15)$$

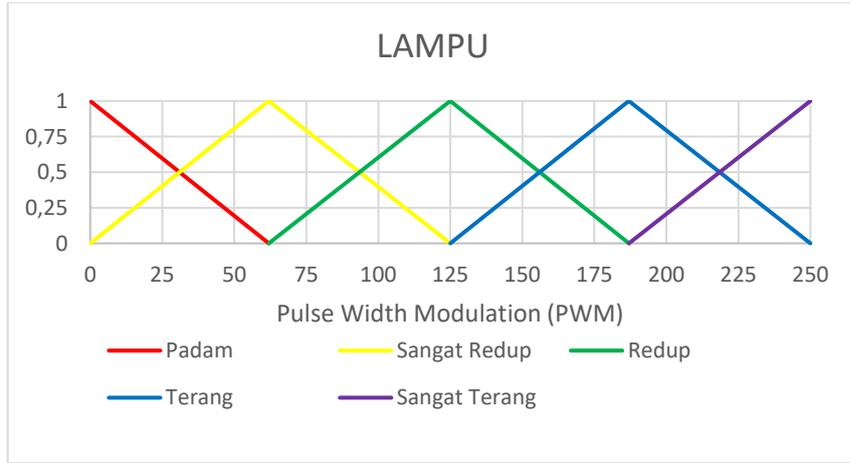
$$\mu_{\text{Sedang}}[x] = \begin{cases} 1, & x \geq 160 \text{ atau } x \leq 180 \\ \frac{x-150}{160-150}, & 150 \leq x \leq 160 \\ \frac{190-x}{190-180}, & 180 \leq x \leq 190 \\ 0, & x \leq 150 \text{ atau } x \geq 190 \end{cases} \quad (5.16)$$

$$\mu_{\text{Cepat}}[x] = \begin{cases} 1, & x \geq 190 \\ \frac{x-180}{190-180}, & 180 \leq x \leq 190 \\ 0, & x \leq 180 \end{cases} \quad (5.17)$$

5.5.1.2 Fungsi keanggotaan *output* lampu

Pada *output* lampu terdapat 5 fungsi keanggotaan yaitu padam, sangat redup, redup, terang dan sangat terang. Masing-masing memiliki *domain range* atau batasan nilai yang mengacu pada penelitian yang dilakukan oleh Irianto, Jaya, dan Anggelo (2011) tentang aplikasi pengkondisian lampu pada ruangan menggunakan logika fuzzy berbasis mikrokontroler. Fungsi keanggotaan *output* lampu dibentuk menggunakan representasi kurva segitiga berdasarkan persamaan 2.3 pada subbab 2.2.1.2. Berikut Gambar 5.10 akan menunjukkan kurva *domain range* atau batasan nilai *output* lampu.





Gambar 5.16 Kurva fungsi keanggotaan output lampu

Berikut Tabel 5.6 akan menunjukkan domain range output lampu.

Tabel 5.17 Domain range output lampu

No.	Fungsi Keanggotaan	Range (PWM)
1	Padam	[0 ≤62]
2	Sangat Redup	[0 125]
3	Redup	[62 187]
4	Terang	[125 250]
5	Sangat Terang	[187 ≥250]

Berikut persamaan fungsi keanggotaan untuk input cahaya.

$$\mu_{\text{Padam}}[x] = \begin{cases} 1, & x \leq 0 \\ \frac{62-x}{62}, & 0 \leq x \leq 62 \\ 0, & x \geq 62 \end{cases} \quad (5.10)$$

$$\mu_{\text{Sangat Redup}}[x] = \begin{cases} \frac{x}{62}, & 0 \leq x \leq 62 \\ \frac{125-x}{125-62}, & 62 \leq x \leq 125 \\ 0, & x \leq 0 \text{ atau } x \geq 125 \end{cases} \quad (5.11)$$

$$\mu_{\text{Redup}}[x] = \begin{cases} \frac{x-62}{125-62}, & 62 \leq x \leq 125 \\ \frac{187-x}{187-125}, & 125 \leq x \leq 187 \\ 0, & x \leq 62 \text{ atau } x \geq 187 \end{cases} \quad (5.12)$$



$$\mu_{\text{Terang}}[X] = \begin{cases} \frac{x-125}{187-125}, & 125 \leq x \leq 187 \\ \frac{250-x}{250-187}, & 187 \leq x \leq 250 \\ 0, & x \leq 125 \text{ atau } x \geq 250 \end{cases} \quad (5.13)$$

$$\mu_{\text{Sangat Terang}}[X] = \begin{cases} 1, & x \geq 250 \\ \frac{x-187}{250-187}, & 187 \leq x \leq 250 \\ 0, & x \leq 187 \end{cases} \quad (5.14)$$

5.5.2 Implementasi Defuzzifikasi pada Arduino

Implementasi defuzzifikasi dilakukan dengan menentukan *domain range* atau batasan nilai menggunakan fungsi *FuzzySet Output* dan *FuzzyOutput object* pada library eFLL mengacu pada subbab 2.2.1.4. Data *domain range* atau batasan nilai pada *output* kipas dan *output* lampu mengacu pada subbab 5.5.1. Berikut Tabel 5.18 akan menunjukkan kode sumber dari *FuzzySet Output* dan *FuzzyOutput object*. Pada baris ke-1 sampai ke-10 adalah kode sumber fungsi keanggotaan *fuzzy output* kipas beserta batasan nilainya, baris ke 12 sampai ke-23 adalah kode sumber fungsi keanggotaan *fuzzy output* lampu beserta batasan nilainya.

Tabel 5.18 Potongan kode sumber fungsi keanggotaan fuzzy output

No	Kode Program
1	FuzzyOutput* kipas = new FuzzyOutput(1);
2	FuzzySet* kpadam = new FuzzySet(0, 0, 0, 0);
3	kipas->addFuzzySet(kpadam);
4	FuzzySet* pelan = new FuzzySet(0, 0, 150, 160);
5	kipas->addFuzzySet(pelan);
6	FuzzySet* ksedang = new FuzzySet(150, 160, 180, 190);
7	kipas->addFuzzySet(ksedang);
8	FuzzySet* cepat = new FuzzySet(180, 190, 250, 250);
9	kipas->addFuzzySet(cepat);
10	fuzzy->addFuzzyOutput(kipas);
11	
12	FuzzyOutput* lampu = new FuzzyOutput(2);
13	FuzzySet* padam = new FuzzySet(0, 0, 0, 62);
14	lampu->addFuzzySet(padam);
15	FuzzySet* sredup = new FuzzySet(0, 62, 62, 125);
16	lampu->addFuzzySet(sredup);
17	FuzzySet* redup = new FuzzySet(62, 125, 125, 187);
18	lampu->addFuzzySet(redup);
19	FuzzySet* lterang = new FuzzySet(125, 187, 187, 250);
20	lampu->addFuzzySet(lterang);
21	FuzzySet* lsterang = new FuzzySet(187, 250, 250, 250);
22	lampu->addFuzzySet(lsterang);
23	fuzzy->addFuzzyOutput(lampu);

Selanjutnya akan dilakukan penggabungan *FuzzyRuleAntecedent* dan *FuzzyRuleConsequent* menjadi satu aturan *fuzzy* dengan menggunakan fungsi *FuzzyRule object*. Berikut Tabel 5.19 akan menunjukkan potongan kode sumber fungsi *FuzzyRule object*. Pada baris ke-1 sampai baris ke-20 adalah kode sumber penggabungan *FuzzyRuleAntecedant* dan *FuzzyRuleConsequent*. Pada baris ke-12

sampai ke-16 adalah kode sumber untuk menambahkan penggabungan *rule* ke dalam *fuzzy*.

Tabel 5.19 Potongan kode sumber fungsi *FuzzyRule* object

No	Kode Sumber
1	<code>FuzzyRule* fuzzyRule1 = new FuzzyRule(1, Rule1,</code>
2	<code>PadamAndPadam);</code>
3	<code>FuzzyRule* fuzzyRule2 = new FuzzyRule(2, Rule2,</code>
4	<code>PadamAndPadam);</code>
5	<code>FuzzyRule* fuzzyRule3 = new FuzzyRule(3, Rule3,</code>
6	<code>PadamAndPadam);</code>
7	<code>FuzzyRule* fuzzyRule4 = new FuzzyRule(4, Rule4,</code>
8	<code>PadamAndPadam);</code>
9	<code>FuzzyRule* fuzzyRule5 = new FuzzyRule(5, Rule5,</code>
10	<code>PadamAndPadam);</code>
11	
12	<code>fuzzy->addFuzzyRule(fuzzyRule1);</code>
13	<code>fuzzy->addFuzzyRule(fuzzyRule2);</code>
14	<code>fuzzy->addFuzzyRule(fuzzyRule3);</code>
15	<code>fuzzy->addFuzzyRule(fuzzyRule4);</code>
16	<code>fuzzy->addFuzzyRule(fuzzyRule5);</code>

Terakhir yaitu mengeksekusi proses defuzzifikasi dengan menggunakan fungsi "*float defuzzify(int id)*", parameter *id* merupakan *Fuzzy* kipas dan *Fuzzy* lampu. Berikut Tabel 5.20 akan menunjukkan kode sumber defuzzifikasi. Pada baris ke-1 merupakan kode sumber untuk mengeksekusi defuzzifikasi *output* kipas dan baris ke-2 adalah kode sumber untuk mengeksekusi defuzzifikasi *output* lampu.

Tabel 5.20 Potongan kode sumber defuzzifikasi

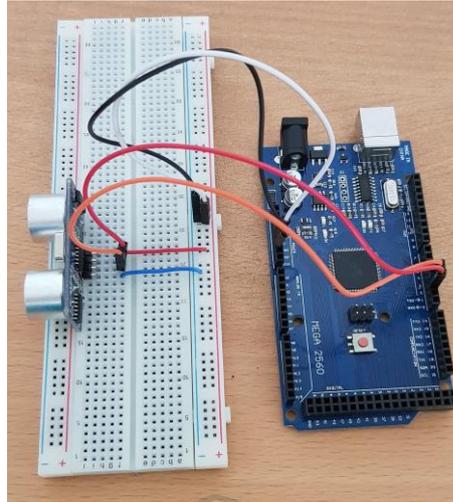
No	Kode Sumber
1	<code>float outKipas = fuzzy->defuzzify(1);</code>
2	<code>float outLampu = fuzzy->defuzzify(2);</code>

5.6 Implementasi Perangkat Sistem dan MATLAB

5.6.1 Implementasi Sensor Gerak

Sensor Ultrasonik untuk gerak akan langsung dihubungkan ke Arduino sesuai dengan pin yang telah dijelaskan pada subbab 5.2.1 yang dapat dilihat pada Gambar 5.17 berikut.





Gambar 5.17 Implementasi *input* sensor gerak

Berikut Tabel 5.21 akan menunjukkan kode sumber pada Arduino IDE yang digunakan untuk membaca *input* pada sensor gerak yang memanfaatkan *library* NewPing, pada kode sumber baris ke-1 merupakan kode sumber untuk membaca *ping sonar* dengan fungsi *.ping* dari sensor ultrasonik, baris ke-2 adalah kode program untuk menghitung jarak (cm) tempuh bolak-balik dari *ping sonar*, dan baris ke-3 sampai baris ke-11 adalah kode sumber pengkondisian untuk pergerakan seseorang dalam ruangan.

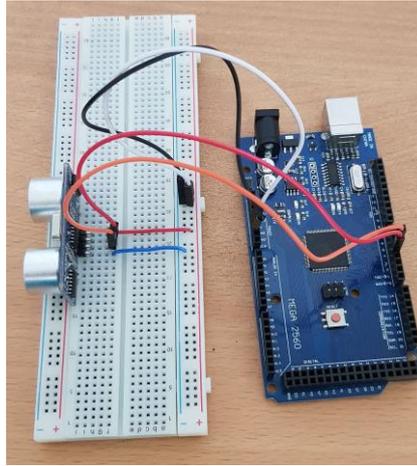
Tabel 5.21 Kode sumber sensor gerak

No	Kode Sumber
1	<code>unsigned int uS = sonar.ping();</code>
2	<code>int ping1 = uS / US_ROUNDTRIP_CM;</code>
3	<code>if(ping1 >= 0 && ping1 <= 200) {</code>
4	<code> Serial.println("Gerak Pasif");</code>
5	<code>}</code>
6	<code>else if(ping1 >= 200 && ping1 <= 300) {</code>
7	<code> Serial.println("Gerak Aktif");</code>
8	<code>}</code>
9	<code>else {</code>
10	<code> Serial.println("Tidak Ada Orang");</code>
11	<code>}</code>

5.6.2 Implementasi Sensor Jarak

Sensor Ultrasonik untuk jarak akan langsung dihubungkan ke Arduino sesuai dengan pin yang telah dijelaskan pada subbab 5.2.2 yang dapat dilihat pada Gambar 5.18 berikut.





Gambar 5.18 Implementasi *input* sensor jarak

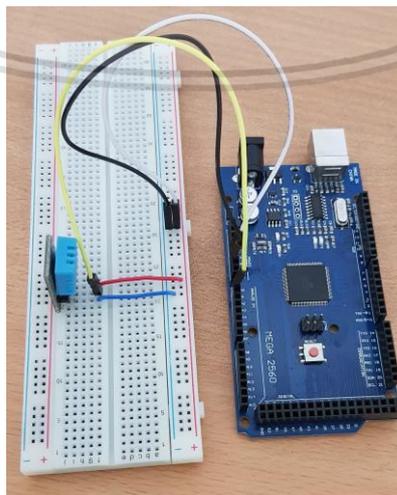
Berikut Tabel 5.22 akan menunjukkan kode sumber yang digunakan untuk membaca jarak (cm) dengan memanfaatkan *library* NewPing. Baris ke-1 merupakan kode sumber untuk membaca *ping sonar* dengan fungsi *.ping* dari sensor ultrasonik. Baris ke-2 adalah kode sumber untuk menghitung jarak (cm) tempuh bolak-balik dari *ping sonar*. Baris ke-3 sampai ke-5 akan mencetak nilai data dari perhitungan jarak.

Tabel 5.22 Kode sumber sensor jarak

No	Kode Sumber
1	<code>unsigned int uS2 = sonar2.ping();</code>
2	<code>int ping2 = uS2 / US_ROUNDTRIP_CM;</code>
3	<code>Serial.print("Jarak: ");</code>
4	<code>Serial.print(ping2);</code>
5	<code>Serial.println(" cm");</code>

5.6.3 Implementasi Sensor Suhu

Pada Gambar 5.19 berikut modul sensor DHT11 dihubungkan ke Arduino sesuai dengan pin yang telah dijelaskan pada subbab 5.2.3.



Gambar 5.19 Implementasi *input* sensor suhu

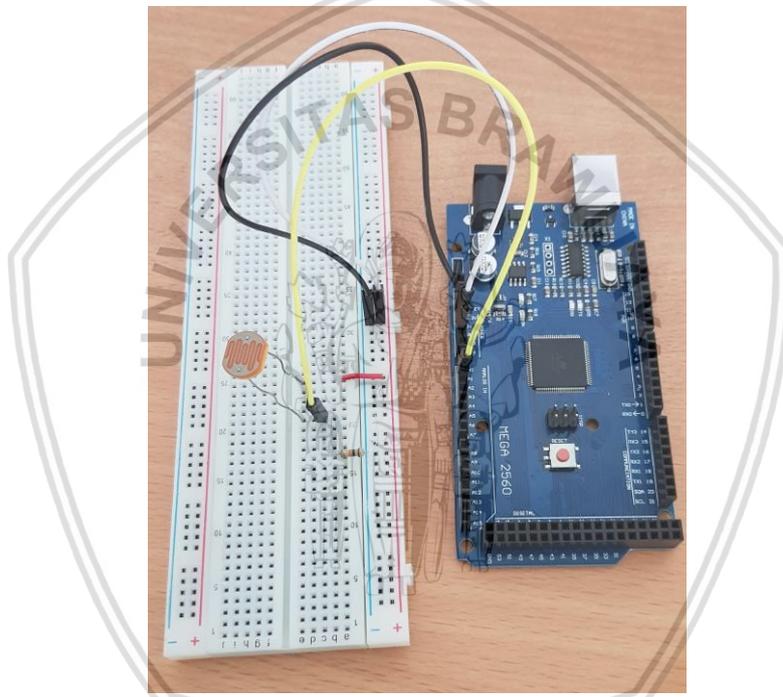
Berikut Tabel 5.23 akan menunjukkan kode sumber yang digunakan untuk membaca suhu dengan memanfaatkan *library* DHT11, fungsi yang digunakan adalah *.temperature* yang dapat dilihat pada baris ke-2.

Tabel 5.23 Kode sumber *input* suhu

No	Kode Sumber
1	<code>int check = DHT.read11(dht_pin);</code>
2	<code>int suhu = DHT.temperature;</code>
3	<code>Serial.print("Suhu = ");</code>
4	<code>Serial.print(suhu);</code>
5	<code>Serial.println("C ");</code>

5.6.4 Implementasi Sensor Cahaya

Pada Gambar 5.20 berikut sensor LDR dihubungkan menggunakan resistor 10k ohm ke Arduino sesuai dengan pin yang telah dijelaskan pada subbab 5.2.4.



Gambar 5.20 Implementasi *input* sensor cahaya

Berikut Tabel 5.24 akan menunjukkan kode sumber yang digunakan untuk membaca *input* sensor cahaya. Pada baris ke-1 akan membaca nilai dari sensor LDR dengan fungsi *analogRead*. Baris ke-4 akan dihitung secara manual untuk mendapatkan nilai lux dari sensor.

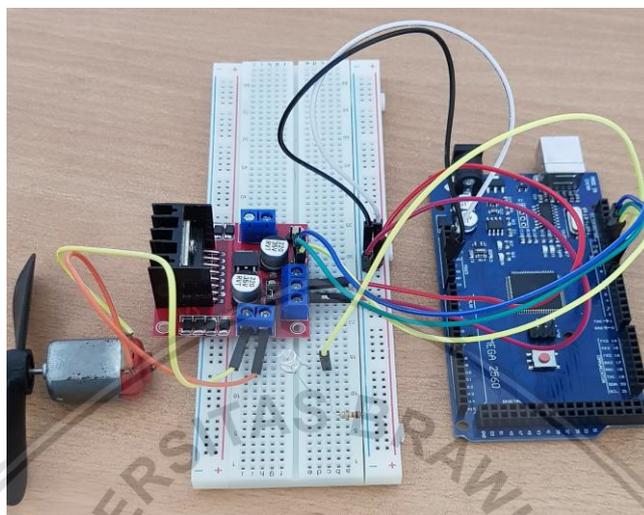
Tabel 5.24 Kode program *input* cahaya

No	Kode Program
1	<code>ldrValue = analogRead(ldrPin);</code>
2	<code>float Vout = (ldrValue * 0.0048828125);</code>
3	<code>lux = 500 / (10*(5 - Vout)/Vout);</code>
4	<code>Serial.print("Cahaya : ");</code>
5	<code>Serial.print(lux);</code>
6	<code>Serial.println(" Lux");</code>



5.6.5 Implementasi Kipas dan Lampu

Implementasi *output* kipas dihubungkan pada driver motor, selanjutnya driver motor dan lampu LED akan dihubungkan ke Arduino sesuai dengan pin yang telah dijelaskan pada subbab 5.2.5. Berikut Gambar 5.21 akan menunjukkan rangkaiannya.



Gambar 5.21 Implementasi *output* kipas dan lampu

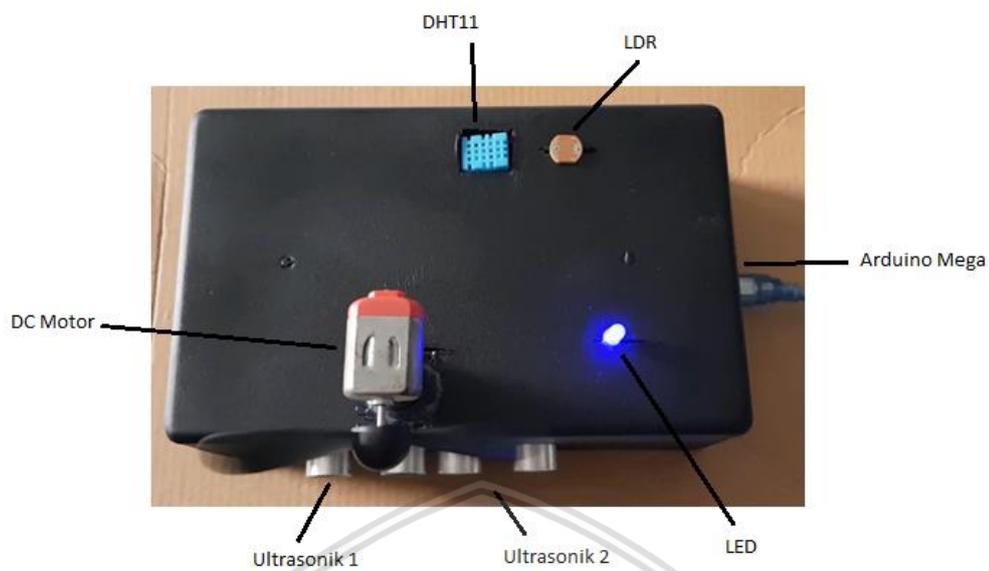
Berikut Tabel 5.25 akan menunjukkan kode sumber untuk menjalankan *output* kipas dan lampu. Pada baris ke-1 sampai ke-3 merupakan kode sumber untuk menjalankan kipas searah dengan jarum jam dan baris ke-5 adalah kode sumber untuk *output* lampu.

Tabel 5.25 Kode sumber output kipas dan lampu

No	Kode Sumber
1	<code>digitalWrite(IN1, LOW);</code>
2	<code>digitalWrite(IN2, HIGH);</code>
3	<code>analogWrite(ENA, outKipas);</code>
4	
5	<code>analogWrite(ledPin, outLampu);</code>

5.6.6 Implementasi Alat Prototipe Sistem

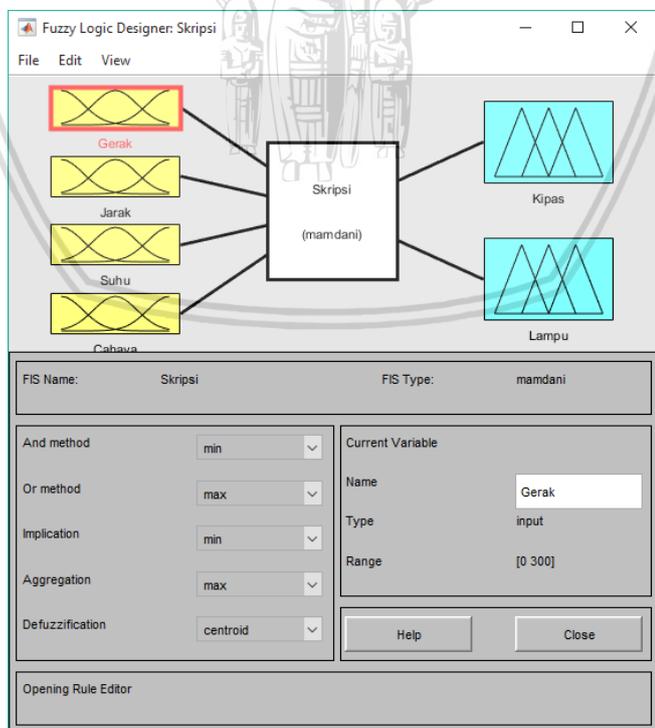
Implementasi alat prototipe sistem dilakukan dengan menghubungkan semua rangkaian yang telah diimplementasikan dan sesuai dengan perancangan prototipe sistem pada subbab 5.2.6. Berikut Gambar 5.22 akan menunjukkan alat prototipe sistem.



Gambar 5.22 Implementasi alat prototipe sistem

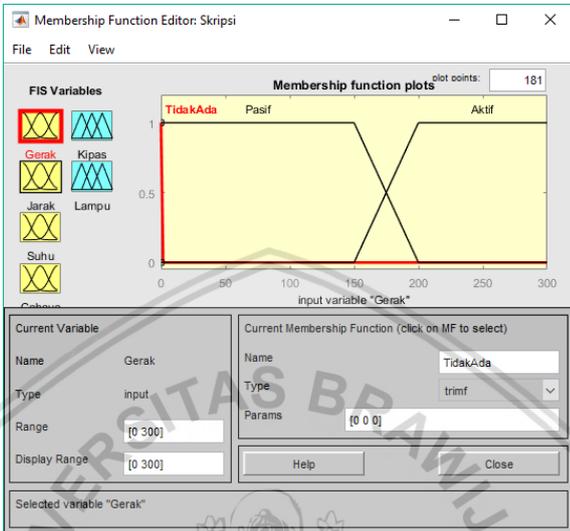
5.6.7 Implementasi MATLAB

Perancangan MATLAB bertujuan untuk melakukan pengujian akurasi output logika *fuzzy* pada sistem dan perhitungan manualnya. Pada MATLAB dibutuhkan sebuah *Fuzzy Logic Toolbox* yang akan memudahkan dalam pemodelan logika *fuzzy* sistem. Berikut Gambar 5.23 akan menunjukkan *Fuzzy Logic Toolbox* pada MATLAB.



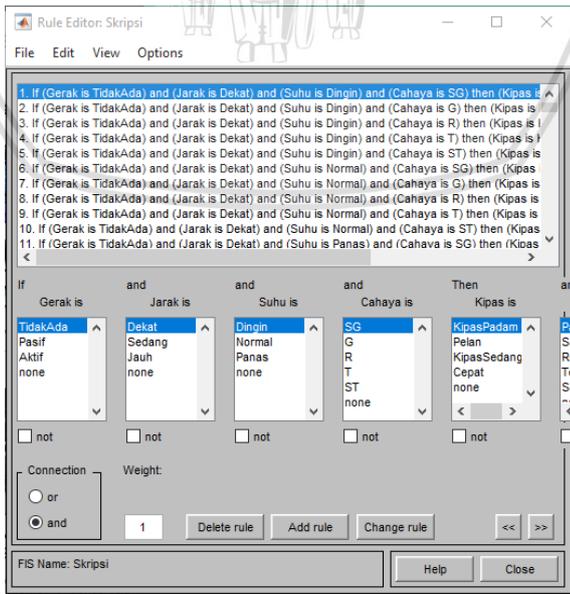
Gambar 5.23 *Fuzzy Logic Toolbox* pada MATLAB

Selanjutnya, akan dibuat fungsi keanggotaan input dan output dengan menggunakan data yang mengacu pada subbab 5.1.1.4 sampai 5.1.1.9. Untuk membuat fungsi keanggotaan pada MATLAB dibutuhkan aplikasi yang disebut *Membership Function Editor*. Aplikasi ini memudahkan dalam pemodelan fungsi keanggotaan baik dari *input* maupun *output*. Berikut Gambar 5.24 akan menunjukkan *Membership Function Editor* pada MATLAB.



Gambar 5.24 Membership Function Editor untuk fungsi keanggotaan

Yang terakhir adalah membuat *rules* atau aturan *fuzzy*. Data aturan *fuzzy* yang digunakan mengacu pada subbab 5.1.1.10. Pada MATLAB akan digunakan aplikasi *Rule Editor*. Aplikasi ini memudahkan untuk melakukan pemodelan kombinasi aturan *input* beserta *output* yang telah dibuat. Berikut Gambar 5.25 akan menunjukkan *Rule Editor* pada MATLAB.



Gambar 5.25 Rule Editor untuk pemodelan aturan fuzzy



BAB 6 PENGUJIAN DAN ANALISIS

Pada bab ini akan dilakukan pengujian untuk mencari akurasi input sensor Ultrasonik gerak dan jarak, DHT11, dan LDR. Nilai *input* sensor nantinya akan dibandingkan dengan nilai *input* yang diambil dengan alat ukur. Selanjutnya dilakukan pengujian kesesuaian *output* sistem dengan rule *fuzzy* atau aturan yang telah dibuat pada subbab 5.1.1.10. Kemudian data dari pengujian kesesuaian *output* sistem akan dibandingkan dengan nilai *output* dari MATLAB yang telah dibuat mengacu pada subbab 5.1.3. Semua pengujian akan dilakukan sebanyak 15 kali. Analisis dilakukan setelah mendapatkan hasil pengujian berupa tabel atau grafik.

6.1 Pengujian Sensor Ultrasonik Gerak dan Jarak

Pengujian sensor Ultrasonik dilakukan untuk mengetahui akurasi dari sensor dalam mendeteksi gerak dan jarak. Ada 2 sensor ultrasonik yang akan diuji, salah satunya digunakan untuk mendeteksi gerak dan yang lainnya untuk mengukur jarak. Nilai *input* dari kedua sensor berupa centimeter (cm) yang akan dibandingkan dengan meteran (cm). Perubahan nilai *input* dilakukan dengan mendeteksi orang pada jangkauan sensor. Berikut persamaan 6.1 merupakan rumus perhitungan akurasi yang akan digunakan.

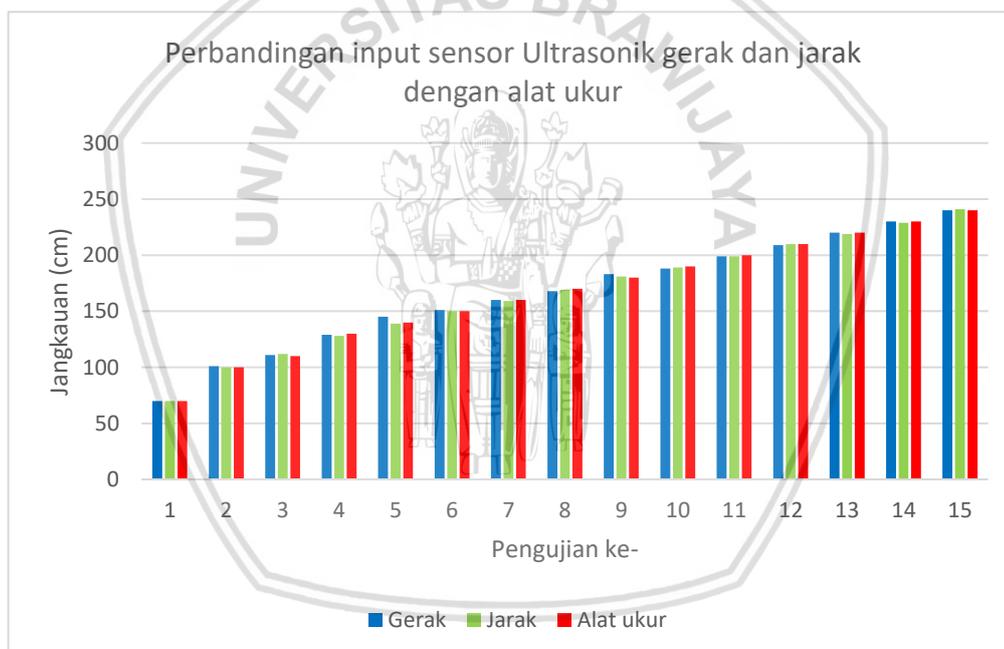
$$\text{Akurasi} = 100\% - \left(\frac{\text{nilai sensor} - \text{nilai alat ukur}}{\text{nilai alat ukur}} \times 100\% \right) \quad (6.1)$$

Berikut Tabel 6.1 akan menunjukkan hasil pengujian perbandingan kedua sensor dengan alat ukur dan Gambar 6.1 akan menunjukkan grafik perbandingannya.

Tabel 6.1 Hasil pengujian akurasi sensor ultrasonik gerak dan jarak

No.	Nilai input Sensor Gerak (cm)	Nilai input Sensor Jarak (cm)	Nilai alat ukur meteran (cm)	Akurasi Sensor Gerak (%)	Akurasi Sensor Jarak (%)
1	70	70	70	100	100
2	101	100	100	99	100
3	111	112	110	99,09	98,18
4	129	128	130	99,23	98,46
5	145	139	140	96,42	99,28
6	151	150	150	99,34	100
7	160	159	160	100	99,37
8	168	169	170	98,82	99,42

No.	Nilai input Sensor Gerak (cm)	Nilai input Sensor Jarak (cm)	Nilai alat ukur meteran (cm)	Akurasi Sensor Gerak (%)	Akurasi Sensor Jarak (%)
9	183	181	180	98,33	99,45
10	188	189	190	98,95	99,47
11	199	199	200	99,5	99,5
12	209	210	210	99,52	100
13	220	219	220	100	99,54
14	230	229	230	100	99,56
15	240	241	240	100	99,58
Rata-rata				99,22	99,45



Gambar 6.1 Grafik perbandingan *input* sensor ultrasonik dengan alat ukur

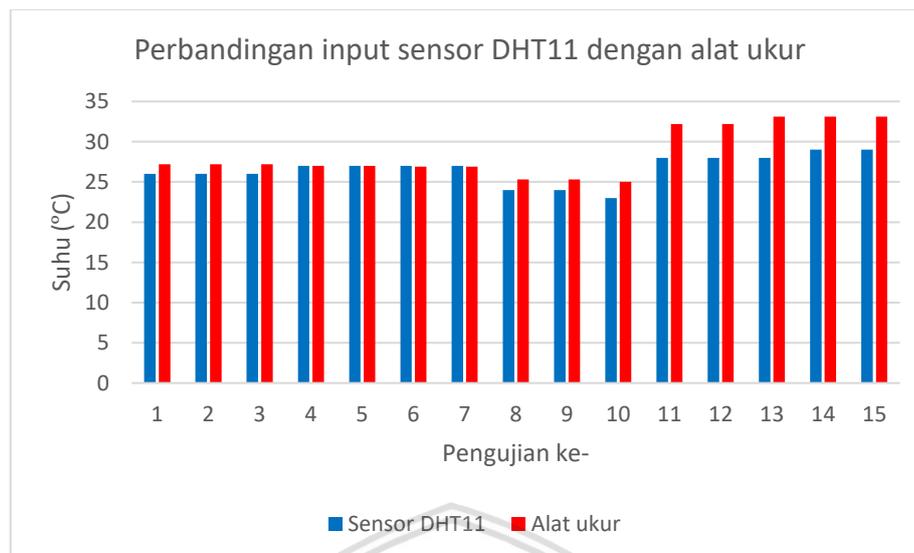
Dari hasil pengujian yang telah dilakukan pada Tabel 6.1 didapatkan rata-rata akurasi untuk gerak sebesar 99,22% dan jarak sebesar 99,45% dari 15 kali pengujian. Dan dapat dilihat pada Gambar 6.1 grafik menunjukkan nilai jangkauan yang tidak jauh berbeda satu sama lain. Dapat disimpulkan bahwa akurasi pembacaan sensor sangat baik dalam mendeteksi adanya keberadaan dan pergerakan seseorang dalam jangkauannya.

6.2 Pengujian Sensor DHT11

Pengujian sensor DHT11 bertujuan untuk mengetahui akurasi dari sensor dalam mendeteksi suhu pada ruangan. Nilai sensor akan dibandingkan dengan nilai alat ukur suhu. Alat ukur yang digunakan adalah HTC-2 *Digital Hygrometer Thermometer Clock*. Perubahan nilai *input* dilakukan dengan menggunakan air panas. Berikut Tabel 6.2 akan menunjukkan hasil perhitungan akurasi dan Gambar 6.2 akan menunjukkan grafik perbandingan nilai sensor dengan alat ukur.

Tabel 6.2 Hasil pengujian akurasi sensor DHT11

No.	Nilai input Sensor (°C)	Nilai input alat ukur (°C)	Akurasi (%)
1	26	27,2	95,58
2	26	27,2	95,58
3	26	27,2	95,58
4	27	27	100
5	27	27	100
6	27	26,9	99,62
7	27	26,9	99,62
8	24	25,3	94,86
9	24	25,3	94,86
10	23	25	92
11	28	32,2	86,95
12	28	32,2	86,95
13	28	33,1	84,59
14	29	33,1	87,61
15	29	33,1	87,61
Rata-rata			93,43



Gambar 6.2 Grafik perbandingan *input* sensor DHT11 dengan alat ukur

Hasil pengujian sensor DHT11 pada Tabel 6.2 didapatkan rata-rata akurasi 93,43% dari 15 kali pengujian. Dapat dikatakan bahwa tingkat akurasi sensor cukup baik dalam mendeteksi suhu di sekitarnya. Namun, dapat dilihat pada Gambar 6.2 pada grafik perbedaan terlihat pada pengujian ke-8 hingga ke-15 saat suhu tinggi akurasi sensor menjadi menurun dari 99,62% pada pengujian ke-7 hingga 84,59% pada pengujian ke-13. Dapat dikatakan bahwa tingkat akurasi sensor DHT11 menjadi menurun ketika suhu tinggi.

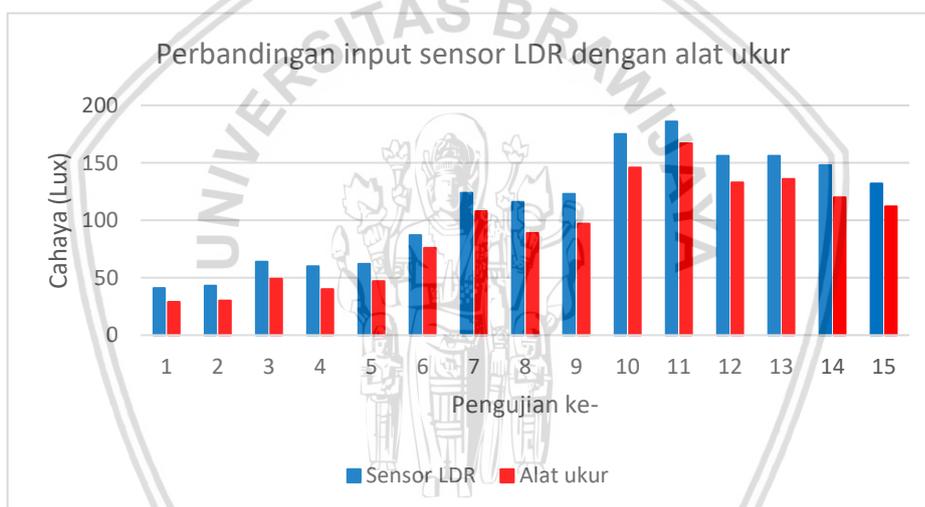
6.3 Pengujian Sensor LDR

Pengujian sensor LDR bertujuan untuk mengetahui akurasi dari sensor dalam mendeteksi cahaya yang masuk pada ruangan. Alat ukur yang digunakan sebagai pembandingan adalah *Digital Lux Light Flux Meter*. Perubahan nilai input dilakukan dengan memberi cahaya dari lampu senter. Berikut Tabel 6.3 akan menunjukkan hasil perhitungan akurasi dan Gambar 6.3 akan menunjukkan grafik perbandingan nilai sensor dengan alat ukur.

Tabel 6.3 Hasil pengujian akurasi sensor LDR

No.	Nilai input Sensor (lux)	Nilai input alat ukur (lux)	Akurasi (%)
1	41	29	58,62
2	43	30	56,67
3	64	49	69,38
4	60	40	50
5	62	47	68,08
6	87	76	85,52
7	124	108	85,18

No.	Nilai input Sensor (lux)	Nilai input alat ukur (lux)	Akurasi (%)
8	116	89	69,66
9	123	97	73,19
10	175	146	80,13
11	186	167	88,62
12	156	133	82,71
13	156	136	85,29
14	148	120	76,67
15	132	112	82,14
Rata-rata			74,12



Gambar 6.3 Grafik perbandingan *input* sensor LDR dengan alat ukur

Dari Tabel 6.3 pengujian sensor LDR didapatkan rata-rata akurasi 74,12% dari 15 kali pengujian. Pada gambar 6.3 grafik memperlihatkan banyak perbedaan hasil pembacaan input sensor dengan alat ukur dikarenakan tingkat akurasi sensor yang rendah dan ketidak-seimbangan pemberian input cahaya yang dihasilkan oleh lampu senter pada sensor dan alat ukur saat pengujian.

6.4 Pengujian Kesesuaian *Output*

Pengujian kesesuaian *output* dilakukan untuk mengetahui kesesuaian *output* alat prototipe sistem dengan rule *fuzzy* atau aturan yang dibuat berdasarkan subbab 5.4.1. Pengujian dilakukan dengan meletakkan alat prototipe sistem pada ruangan yang nantinya akan mendeteksi suhu ruangan dan cahaya yang masuk pada ruangan. Untuk pergerakan akan dilakukan secara pasif atau dalam keadaan duduk diam dalam jangkauan sensor dari alat prototipe sistem dan pengujian dilakukan selama 24 jam dengan pengambilan nilai berdasarkan jam/pukul

tertentu. Berikut Tabel 6.4 akan menunjukkan hasil pengujian kesesuaian *output* kipas dan lampu dari alat prototipe.

Tabel 6.4 Hasil pengujian kesesuaian *output* sistem

No.	Pukul	Nilai Input Sensor				Nilai Output Alat Prototipe		Kesesuaian dengan Rule
		Gerak (cm)	Jarak (cm)	Suhu (°C)	Cahaya (lux)	Kipas (PWM)	Lampu (PWM)	
1	06:00	75	75	23	192	111,24	168,59	Sesuai
2	07:30	75	76	23	236	111,24	142,67	Sesuai
3	09:00	74	75	23	447	111,24	46,97	Sesuai
4	10:30	74	75	24	559	132,5	22,91	Sesuai
5	12:00	75	75	24	559	132,5	22,91	Sesuai
6	13:30	70	70	24	563	132,5	22,91	Sesuai
7	15:00	70	71	24	432	128,25	51,07	Sesuai
8	16:30	70	70	24	254	132,5	118,29	Sesuai
9	18:00	70	70	24	49	132,5	228,3	Sesuai
10	19:30	74	75	24	13	132,5	228,3	Sesuai
11	21:00	74	75	23	11	111,24	226,6	Sesuai
12	22:30	75	75	23	11	96,71	224,71	Sesuai
13	24:00	75	75	22	10	96,71	224,71	Sesuai
14	01:30	75	75	22	11	96,71	224,71	Sesuai
15	03:00	74	75	22	10	96,71	224,71	Sesuai

Dari pengujian yang telah dilakukan pada Tabel 6.4 bahwa semua *output* yang dihasilkan dari 15 kali pengujian semuanya sesuai dengan rule fuzzy atau aturan yang dibuat. Ini menunjukkan bahwa sistem dapat berjalan dengan sangat baik dalam mendeteksi *input* dan menghasilkan *output* yang sesuai dengan rule fuzzy atau aturan yang telah dibuat.

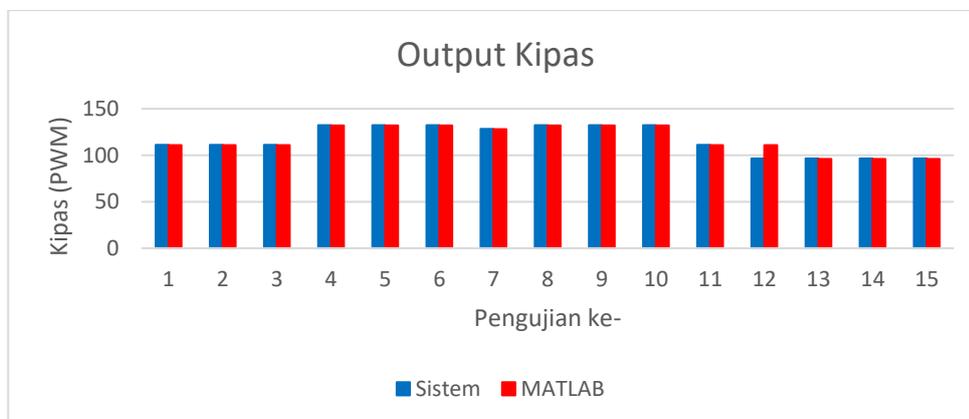
6.5 Pengujian Logika Fuzzy dengan MATLAB

Pengujian logika fuzzy dilakukan dengan membandingkan nilai output sistem yang telah di dapat dari kesesuaian *output* pada subbab 6.4 dengan *output* yang di dapat dari MATLAB mengacu pada subbab 5.1.3 dan selanjutnya akan dilakukan perhitungan akurasi, pada pengujian hasil output sistem dan MATLAB menggunakan tipe data *floating point*. Berikut Tabel 6.5 akan menunjukkan hasil pengujian akurasinya.

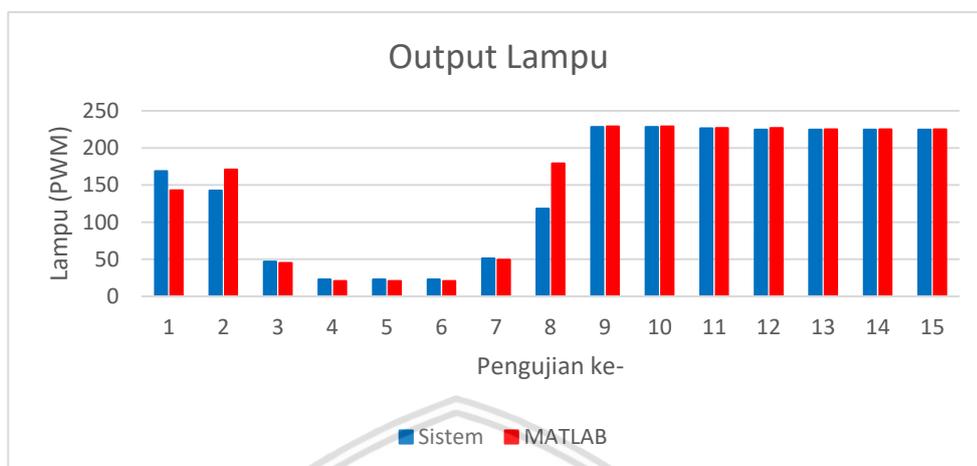
Tabel 6.5 Hasil pengujian akurasi *output* sistem dengan *output* MATLAB

No.	Nilai Output Sistem		Nilai Output MATLAB		Akurasi (%)	
	Kipas (PWM)	Lampu (PWM)	Kipas	Lampu	Kipas	Lampu
1	111,24	168,59	111	143	99,78	82,1
2	111,24	142,67	111	171	99,78	83,43
3	111,24	46,97	111	45	99,78	95,63
4	132,5	22,91	132	20,6	99,62	88,79
5	132,5	22,91	132	20,6	99,62	88,79
6	132,5	22,91	132	20,6	99,62	88,79
7	128,25	51,07	128	49,4	99,8	96,62
8	132,5	118,29	132	179	99,62	66,08
9	132,5	228,3	132	229	99,62	99,69
10	132,5	228,3	132	229	99,62	99,69
11	111,24	226,6	111	227	99,78	99,82
12	96,71	224,71	111	227	87,13	98,99
13	96,71	224,71	96	225	99,26	99,87
14	96,71	224,71	96	225	99,26	99,87
15	96,71	224,71	96	225	99,26	99,87
Rata-rata					98,77	92,54

Berikut Gambar 6.4 dan Gambar 6.5 akan menunjukkan grafik perbandingan nilai output sistem dengan nilai *output* matlab.

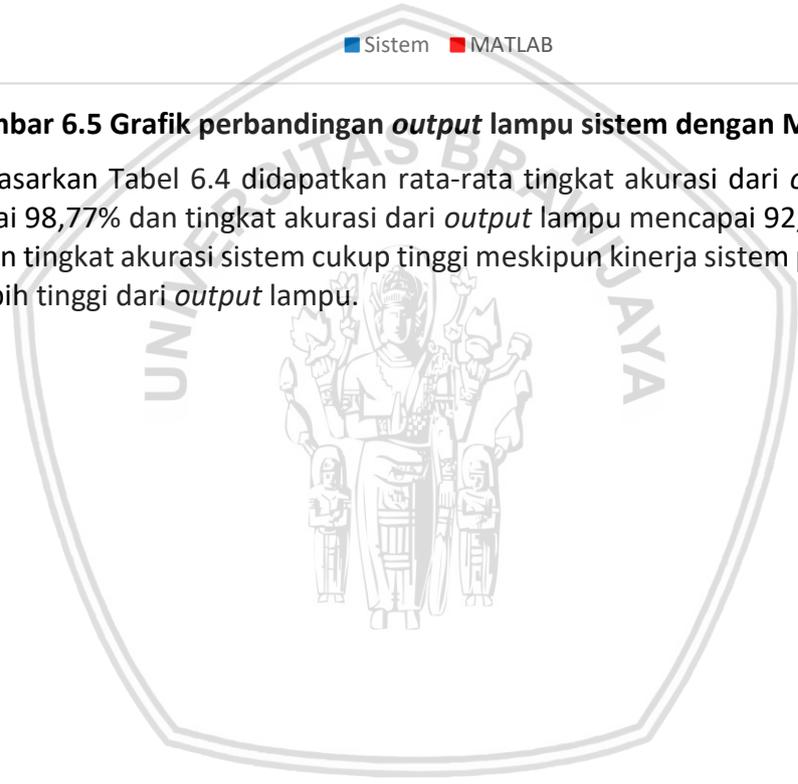


Gambar 6.4 Grafik perbandingan *output* kipas sistem dengan MATLAB



Gambar 6.5 Grafik perbandingan *output* lampu sistem dengan MATLAB

Berdasarkan Tabel 6.4 didapatkan rata-rata tingkat akurasi dari *output* kipas mencapai 98,77% dan tingkat akurasi dari *output* lampu mencapai 92,54%. Dapat dikatakan tingkat akurasi sistem cukup tinggi meskipun kinerja sistem pada *output* kipas lebih tinggi dari *output* lampu.



BAB 7 KESIMPULAN DAN SARAN

7.1 Kesimpulan

Berdasarkan rumusan masalah pada penelitian, hasil implementasi, pengujian dan analisis yang telah dilakukan dalam penelitian ini, maka dapat ditarik kesimpulan sebagai berikut.

1. Untuk membuat sistem dibutuhkan beberapa perangkat sebagai *input*, proses dan *output*. Pada bagian input dibutuhkan 2 sensor ultrasonik sebagai pendeteksi gerak dan jarak, sensor DHT11 sebagai pendeteksi suhu pada ruangan, dan sensor LDR sebagai pendeteksi cahaya yang masuk pada ruangan. Pada bagian proses akan digunakan mikrokontroler Arduino Mega. Yang terakhir bagian *output* akan digunakan DC motor sebagai kipas dan LED sebagai lampu. Dalam merancang logika *fuzzy* diperlukan 3 tahap, tahap pertama fuzzifikasi yaitu menentukan fungsi keanggotaan dari *input* dan *output* dengan melakukan pemetaan data tegas ke himpunan *fuzzy* pada tahap ini ada 4 variabel input yang digunakan antara lain gerak, jarak, suhu dan cahaya kemudian untuk variabel output ada 2 yaitu kipas dan lampu, tahap kedua inferensi yaitu pembentukan *rule* dan mencari *rule* evaluation dan *rule* aggregation pada tahap ini dibuat sebanyak 135 *rule*, sistem menggunakan metode implikasi MIN pada library, defuzzifikasi yaitu mengkonversi semua nilai *crisp* menjadi nilai tegas, sistem menggunakan metode COA (*Center Of Area*) pada library.
2. Untuk implementasi logika *fuzzy* digunakan Arduino Mega sebagai mikrokontroler sistem, selanjutnya digunakan fungsi-fungsi pada library eFLL untuk mempermudah implementasi logika *fuzzy* pada Arduino.
3. Tingkat akurasi sistem pada penelitian didapatkan dari perbandingan nilai *output* sistem dengan *output* MATLAB. Rata-rata akurasi yang didapatkan dari *output* kipas yaitu 98,77% dan *output* lampu yaitu 92,54%, ini menunjukkan bahwa kinerja dan akurasi dari *output* sistem yang dibuat cukup tinggi. Sedangkan untuk rata-rata akurasi dari tiap *input* sensor yang didapat yaitu 99,22% untuk gerak, 99,45% untuk jarak, 93,43% untuk suhu, dan 74,12% untuk cahaya.

7.2 Saran

Berikut beberapa saran dari penulis untuk pengembangan lebih lanjut dari penelitian ini.

1. Dapat menambahkan input sensor, *output* dan fungsi keanggotaan pada sistem.
2. Mengganti sensor cahaya yang memiliki akurasi yang lebih tinggi.
3. Dapat menggunakan metode lain pada logika *fuzzy*.

DAFTAR PUSTAKA

- Alves, A. & Lemos, M. M., 2016. *eFLL (Embedded Fuzzy Logic Library)*. [Online] Available at: <https://github.com/zerokol/eFLL> [Diakses 19 12 2017].
- Arduino, t.thn. *ArduinoBoardMega*. [Online] Available at: <https://www.arduino.cc/en/Main/arduinoBoardMega> [Diakses 4 September 2017].
- Badan Pengkajian dan Penerapan Teknologi, 2015. *Outlook Energi Indonesia (OEI) 2015*. Jakarta: Pusat Teknologi Pengembangan Sumberdaya Energi.
- Budiharto, W. & Suhartono, D., 2014. *Artificial Intelligence : Konsep dan Penerapannya*. Jakarta: CV. Andi Offset.
- Caldo, R. B., 2015. *Design and Development of Fuzzy Logic Controlled Dimming Lighting System Using Arduino Microcontroller*. Philippines, 8th IEEE International Conference Humanoid.
- Elec Freaks, t.thn. *Ultrasonic Ranging Module HC - SR04 datasheet*. [Online] Available at: <https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf> [Diakses 19 12 2017].
- Irianto, Jaya, A. & Anggelo, D., 2011. *Application of Fuzzy Logic to Control Room Illumination Based Microcontroller*. Surabaya, Politeknik Elektronika Negeri Surabaya.
- Kusumadewi, S. & Purnomo, H., 2013. *Aplikasi Logika Fuzzy Untuk Pendukung Keputusan*. 2nd penyunt. Yogyakarta: Graha Ilmu.
- Mulia, N. A., 2017. *Implementasi Logika Fuzzy Untuk Purwarupa Pengkondisian AC dan Lampu Otomatis pada Sebuah Ruangan*, s.l.: Universitas Brawijaya.
- Oltean, G. & Ivanciu, L.-N., 2017. *Implementation of a Fuzzy Logic-Based Embedded System for Temperature Control*. s.l., IEEE Xplore Digital Library.
- Purwaningrum, N., 2007. *Aplikasi Fuzzy Logic untuk Pengendali Penerangan Ruangan Berbasis Mikrokontroler ATmega8535*, Semarang: Teknik Elektro Fakultas Teknik Universitas Semarang.
- Saelan, A., 2009. *Logika Fuzzy*, Bandung: Institut Teknologi Bandung.
- STMicroelectronics, t.thn. *L298 Dual Full-Bridge Drive datasheet*. [Online] Available at: <https://www.sparkfun.com/datasheets/Robotics/L298 H Bridge.pdf> [Diakses 19 December 2017].
- Sunrom Technologies, 2012. *DHT11 - Humidity and Temperature Sensor*. [Online] Available at: www.sunrom.com/p-1141.html [Diakses 16 September 2017].

Susilo, F., 2006. *Himpunan dan logika kabur serta aplikasinya*. Yogyakarta: Graha Ilmu.

Tarigan, P., 2013. *Sistem Pengendali Pendingin Ruangan Menggunakan Fuzzy Logic Berbasis Mikrokontroler ATmega 8535*, Medan: Informasi dan Teknologi Ilmiah (INTI).

Tarigan, P., Sinurat, S. & Sinambela, M., 2015. *Implementation of A Mamdani Fuzzy Logic Controller for Building Automation Using Electronic Control Based on AT89S51*. s.l., IEEE Xplore Digital Library.

Zaki, D. M., 2016. *Otomatisasi Terang Lampu pada Smart Home dalam Rangka Efisiensi Energi*, Malang: Universitas Brawijaya.

