



Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- Pengutipan tidak merugikan kepentingan yang wajar Unand.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Unand.

IMPLEMENTASI FUZZY LOGIC DAN SENSOR WARNA PADA MESIN PENGADUK KOPI OTOMATIS

TUGAS AKHIR



**ILFIYANTRI INTYAS
07175049**

**JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS ANDALAS
PADANG 2012**

KATA PENGANTAR



Alhamdulillahirabbil'alamiin, dengan mengucapkan syukur kepada Allah SWT yang telah memberikan petunjuk, rahmat, hidayah, dan segala nikmat-Nya kepada seluruh hamba-Nya. Dan tak lupa shalawat serta salam dihaturkan kepada Rasulullah Muhammad SAW, sehingga penulis dapat menyelesaikan penelitian tugas akhir yang berjudul **"IMPLEMENTASI FUZZY LOGIC DAN SENSOR WARNA PADA PENGADUK KOPI OTOMATIS"**, yang bertujuan untuk memperoleh gelar Sarjana Teknik pada Fakultas Teknik Universitas Andalas.

Dalam penulisan tugas akhir ini, tidak lepas dari dukungan dan bantuan berbagai pihak, pada kesempatan ini penulis mengucapkan rasa terima kasih kepada :

1. Kedua orang tua tercinta (H. Indrawan Yulda Mathias dan Elmita), Tante Erliwati dan segenap keluarga (Imam Nanda Intyas S,Di. dan Ikhsan Hariko Intyas A.MD.) yang telah memberikan kasih sayang, dukungan moril dan materil serta doa yang senantiasa menyertai penulis selama menempuh pendidikan ini. Semoga Allah selalu memberikan kasih sayang yang berlipat ganda kepada keluarga kita.
2. Bapak Dr. Eng. Rahmadi Kurnia, selaku Ketua Jurusan, Kepala Laboratorium Telekomunikasi, Kepala Laboratorium Sentral Telepon Otomat Teknik Elektro Fakultas Teknik Universitas Andalas dan sekaligus sebagai pembimbing tugas akhir atas segala bantuan berupa ilmu, bimbingan dan kesabaran yang telah diberikan selama penulis melakukan penelitian.
3. Bapak Ir. Darwison, MT., Ibu Fitrilina, MT., Bapak Heru Dibyo Laksono, MT., dan Ibu Meza Silvana, ST., yang telah bersedia menjadi penguji dari

seminar proposal hingga sidang tugas akhir. Terimakasih atas saran dan masukannya yang berharga untuk kesempurnaan penelitian ini.

4. Bapak dan Ibu Dosen di lingkungan Fakultas Teknik umumnya dan Jurusan Teknik Elektro khususnya atas segala didikan dan ilmu yang telah diberikan.
5. Pegawai Jurusan Pak Eka, Pak Akbar, Bu Gus, Bu Neng, Ni Desi, Ni Yuli dan Refni atas bantuan dan kemudahan dalam segala urusan administrasi.
6. Teman-teman tercinta yang selalu setia menemani, memberikan dukungan dan masukan serta semangat kepada penulis dalam menyelesaikan penelitian ini.
7. Serta kepada semua pihak yang telah membantu penulis dalam menyelesaikan penelitian ini.

Akhir kata hanya kepada Allah SWT tempat berserah diri, semoga apa yang telah kita lakukan diterima sebagai amal yang baik oleh Allah SWT dan bermanfaat bagi kita semua.

Padang, 10 Mei 2012

Penulis



KATA PENGANTAR



Alhamdulillahirabbil'alamiin, dengan mengucapkan syukur kepada Allah SWT yang telah memberikan petunjuk, rahmat, hidayah, dan segala nikmat-Nya kepada seluruh hamba-Nya. Dan tak lupa shalawat serta salam dihaturkan kepada Rasulullah Muhammad SAW, sehingga penulis dapat menyelesaikan penelitian tugas akhir yang berjudul **"IMPLEMENTASI FUZZY LOGIC DAN SENSOR WARNA PADA PENGADUK KOPI OTOMATIS"**, yang bertujuan untuk memperoleh gelar Sarjana Teknik pada Fakultas Teknik Universitas Andalas.

Dalam penulisan tugas akhir ini, tidak lepas dari dukungan dan bantuan berbagai pihak, pada kesempatan ini penulis mengucapkan rasa terima kasih kepada :

1. Kedua orang tua tercinta (H. Indrawan Yulda Mathias dan Elmita), Tante Eriwati dan segenap keluarga (Imam Nanda Intyas S,Di. dan Ikhsan Hariko Intyas A.MD.) yang telah memberikan kasih sayang, dukungan moril dan materil serta doa yang senantiasa menyertai penulis selama menempuh pendidikan ini. Semoga Allah selalu memberikan kasih sayang yang berlipat ganda kepada keluarga kita.
2. Bapak Dr. Eng. Rahmadi Kurnia, selaku Ketua Jurusan, Kepala Laboratorium Telekomunikasi, Kepala Laboratorium Sentral Telepon Otomat Teknik Elektro Fakultas Teknik Universitas Andalas dan sekaligus sebagai pembimbing tugas akhir atas segala bantuan berupa ilmu, bimbingan dan kesabaran yang telah diberikan selama penulis melakukan penelitian.
3. Bapak Ir. Darwison, MT., Ibu Fitrilina, MT., Bapak Heru Dibyo Laksono, MT., dan Ibu Meza Silvana, ST., yang telah bersedia menjadi penguji dari

seminar proposal hingga sidang tugas akhir. Terimakasih atas saran dan masukannya yang berharga untuk kesempurnaan penelitian ini.

4. Bapak dan Ibu Dosen di lingkungan Fakultas Teknik umumnya dan Jurusan Teknik Elektro khususnya atas segala didikan dan ilmu yang telah diberikan.
5. Pegawai Jurusan Pak Eka, Pak Akbar, Bu Gus, Bu Neng, Ni Desi, Ni Yuli dan Refni atas bantuan dan kemudahan dalam segala urusan administrasi.
6. Teman-teman tercinta yang selalu setia menemani, memberikan dukungan dan masukan serta semangat kepada penulis dalam menyelesaikan penelitian ini.
7. Serta kepada semua pihak yang telah membantu penulis dalam menyelesaikan penelitian ini.

Akhir kata hanya kepada Allah SWT tempat berserah diri, semoga apa yang telah kita lakukan diterima sebagai amal yang baik oleh Allah SWT dan bermanfaat bagi kita semua.

Padang, 10 Mei 2012

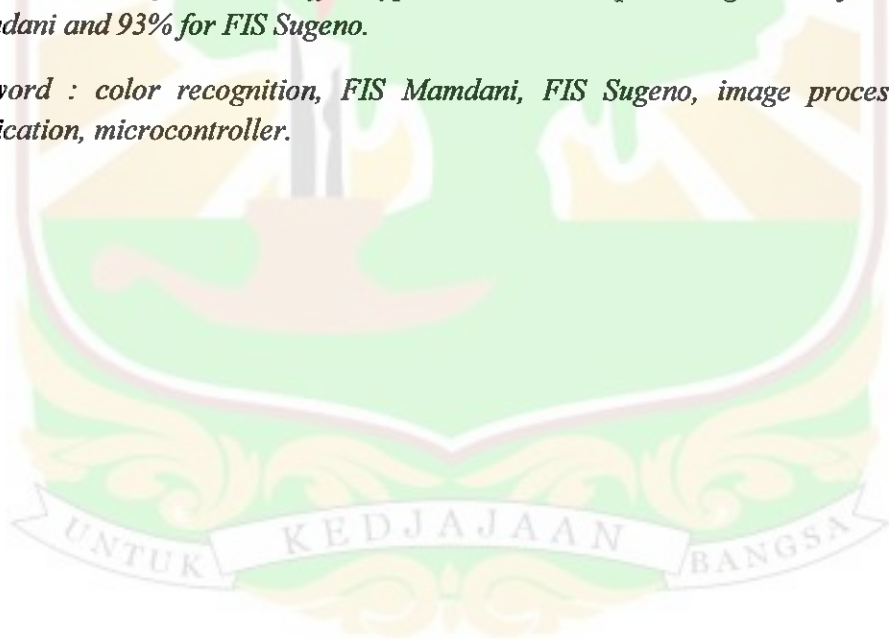
Penulis



ABSTRACT

The rapid development of technology is linked to the complex human need, one of the examples is the implementation of image processing technology to run an application. In this final submission, an image processing system is developed to run a color-based automatic coffee mixer. First, a user chooses the wanted coffee type by clicking one of the buttons provided in GUI on computer screen, and then the computer will send data to the interface system. Based on the data, interface system will run the dc motor and stepper motor. A camera will capture the image of the coffee that is being mixed in a glass. From the coffee image, the input for the FIS system will be obtained. FIS system will determine the coffee type in the glass. If the coffee in the glass matches the coffee type that the user chose in the beginning, the computer will send data to the interface system and the interface system will stop the rotation of dc motor and stepper motor. If the coffee in the glass has not match the coffee type that the user chose, computer will not send any data to the interface system and will order the camera to capture another image of the coffee. To control the coffee mixer, ATMEGA8535 is used. The coffee mixer works automatically according to the input of the computer. The decision maker system is using Fuzzy Inference System FIS). This system has been tested 20 times for each coffee type with success percentage 92% for FIS Mamdani and 93% for FIS Sugeno.

Keyword : color recognition, FIS Mamdani, FIS Sugeno, image processing application, microcontroller.



ABSTRAK

Perkembangan teknologi yang semakin pesat berkaitan dengan semakin kompleksnya kebutuhan manusia, salah satunya adalah penerapan teknologi pengolahan citra dalam menjalankan suatu aplikasi. Pada tugas akhir ini, dikembangkan sistem pengolahan citra untuk menjalankan alat pengaduk kopi otomatis berdasarkan warna. Pertama-tama user akan memilih jenis kopi yang diinginkan dengan mengklik salah satu tombol pada GUI kemudian komputer akan mengirimkan data ke antarmuka. Berdasarkan data tersebut, antarmuka akan menggerakkan motor dc dan stepper. Kamera akan meng-capture gambar kopi yang sedang diaduk. Dari citra kopi tersebut diperoleh input untuk sistem FIS. FIS akan memutuskan jenis kopi dalam gelas. Apabila kopi telah cocok dengan pilihan user, komputer akan mengirimkan data ke antarmuka dan antarmuka akan menghentikan motor dc dan stepper. Apabila kopi belum cocok dengan pilihan user, program akan kembali memerintahkan kamera untuk meng-capture kembali gambar kopi. Pengendalian alat pengaduk kopi otomatis menggunakan mikrokontroler ATMEGA8535. Alat pengaduk kopi otomatis bekerja berdasarkan input dari komputer. Sistem pengambilan keputusan pada sistem menggunakan Fuzzy Inference System (FIS). Sistem ini sudah diujikan sebanyak 20 kali untuk setiap jenis kopi yang diujikan dengan persentase keberhasilan 92% untuk FIS Mamdani, dan 93% untuk FIS Sugeno.

Kata kunci: Pengenalan warna, FIS Mamdani, FIS Sugeno, aplikasi pengolahan citra, mikrokontroler.

DAFTAR ISI

ABSTRAK	i
DAFTAR ISI.....	ii
DAFTAR GAMBAR	v
DAFTAR TABEL.....	vii
DAFTAR ISTILAH DAN SINGKATAN	viii
DAFTAR LAMPIRAN.....	x
BAB I. PENDAHULUAN	
1.1 Latar Belakang.....	1
1.2 Tujuan Penelitian.....	3
1.3 Manfaat Penelitian.....	3
1.4 Batasan Masalah.....	3
1.5 Metodologi penelitian	4
1.6 Sistematika Penulisan	4
1.7 Desain mesin	6
BAB II. TINJAUAN PUSTAKA	
2.1 Logika Fuzzy	7
2.2 Penerapan logika fuzzy	7
2.2.1 Fuzzifikasi.....	9
2.2.2 Evaluasi <i>rule</i> (<i>rule evaluation</i>)	10
2.2.3 Defuzzifikasi	11
2.3 Video	12
2.3.1 Pengertian video.....	12
2.3.2 Pengambilan <i>frame</i> pada video	14
2.3.3 Pengolahan citra dengan logika fuzzy	14

2.3.4 Histogram citra.....	15
2.3.5 Sistem komponen ruang warna	16
2.3.5.1 RGB	16
2.3.5.2 HSI (<i>Hue Saturation Intensity</i>) / HSV / HLS.....	18
2.3.5.4 Kopi Pekat/ <i>Black</i> , Biasa/ <i>Medium</i> , dan Ringan/ <i>Light</i>	20
2.4 Antarmuka	21
2.4.1 Mikrokontroler ATMEGA 8535	21
2.4.1.1 Konfigurasi Mikrokontroler ATMEGA 8535	21
2.4.1.2 Sistem Minimum Mikrokontroler ATMEGA 8535.....	23
2.4.1.3 Sistem Clock.....	25
2.4.1.4 Organisasi Memori AVR ATMega8535	26
2.4.2 Motor DC.....	30
2.4.3 Motor Stepper	31

BAB III METODELOGI PENELITIAN

3.1 Jenis Penelitian	35
3.2 Sampel Penelitian	35
3.3 Teknik Analisa Sistem	36
3.4 Prosedur Penelitian	37

BAB IV DESAIN SISTEM

4.1 Desain Sistem	40
4.2 Perancangan perangkat lunak (<i>software</i>).	40
4.2.1 Membuat video objek (akuisisi data)	40
4.2.2 Konversi Ruang Warna Citra.....	42
4.2.3 Penggunaan <i>Fuzzy Logic Toolbox</i> dalam Pembentukan <i>Database</i> Jenis Kopi pada Sistem Penarik Kesimpulan.....	42
4.2.3.1 <i>FIS Editor</i>	43
4.2.3.2 <i>Membership Function Editor</i>	46

4.2.3.3 <i>Rule Editor</i>	48
4.2.3.4 <i>Defuzzifikasi</i>	49
4.2.4 Perancangan <i>form</i> GUI Otomatisasi Pergerakan Pengaduk Kopi	51
4.3 <i>I/O Interface</i>	53
4.3.1 Port Paralel	53
4.3.2 Mikrokontroller ATMEGA 8535.....	54
4.3.3 Motor DC.....	55
4.3.4 Motor Stepper	55
4.4 Evaluasi Terhadap Hasil Penelitian	56
BAB V ANALISA DAN PEMBAHASAN	
5.1 Hasil <i>capture</i>	58
5.2 Pengujian <i>Fuzzy Inference System</i> dan Kinerja Mesin Pengaduk Kopi.	60
5.2.1 <i>Membership function</i>	60
5.2.2 <i>Rule</i> yang Digunakan	63
5.2.3 Hasil Pengadukan Kopi DenganMesin Kopi Otomatis.....	64
5.2.3.1 Hasil Pengujian dengan menggunakan FIS Mamdani	64
5.2.3.2 Hasil pengujian dengan FIS Sugeno	73
5.2.3.3 Analisa hasil pengujian FIS Mamdani dan FIS Sugeno.....	81
BAB VI PENUTUP	
6.1 Kesimpulan.....	83
6.2 Saran	83
DAFTAR PUSTAKA	

DAFTAR GAMBAR

Gambar 1.1	Desain Mesin pengaduk kopi otomatis.....	6
Gambar 2.1	Konsep dari Logika Fuzzy.....	8
Gambar 2.2	Bagian-Bagian Fungsi Keanggotaan	8
Gambar 2.3	Fungsi Keanggotaan	10
Gambar 2.4	Flow Diagram Proses Fuzzifikasi.....	10
Gambar 2.5	Metode <i>Center of Gravity</i>	11
Gambar 2.6	Metoda <i>Center of Gravity</i> pada <i>Singleton</i>	12
Gambar 2.7	Komponen Video	13
Gambar 2.8	<i>Single Frame</i> yang dirangkai menjadi Video	13
Gambar 2.9	Proses Penangkapan <i>Image</i> Objek Pada Kamera.....	14
Gambar 2.10	Ruang warna RGB	17
Gambar 2.11	HSI <i>color space</i>	19
Gambar 2.12	Konfigurasi Pin-pin Mikrokontroler ATMEGA 8535.....	21
Gambar 2.13	Arsitektur Mikrokontroler ATMEGA 8535.....	23
Gambar 2.14	Sistem Minimum mikrokontroler AT Mega 8535	25
Gambar 2.15	Sistem <i>Clock</i>	26
Gambar 2.16	Konfigurasi Data AVR AT Mega 8535	27
Gambar 2.17	Peta Memori Program AT Mega 8535	28
Gambar 2.18	Bagian Motor Arus Searah.....	30
Gambar 2.19	Penampang Melintang Dari Motor Stepper Tipe Variable Reluctance (VR).....	33
Gambar 2.20	Motor Stepper Tipe Permanent Magnet	33
Gambar 2.21	Penampang Melintang Dari Motor Stepper Tipe Hybrid	34

Gambar 4.1	Blok Diagram Sistem.....	40
Gambar 4.2	<i>Webcam</i>	42
Gambar 4.3	<i>FIS Editor</i>	44
Gambar 4.4	Diagram FIS <i>coffee</i>	46
Gambar 4.5	Diagram FIS <i>coffeesugeno</i>	46
Gambar 4.6	<i>Membership Function Editor</i>	47
Gambar 4.7	Hasil Pembentukan Fungsi Keanggotaan Melalui <i>Membership Function editor</i>	49
Gambar 4.8	Pembentukan <i>Rule</i> Melalui <i>Rule Editor</i>	50
Gambar 4.9	<i>Rule viewer</i> FIS Mamdani.....	51
Gambar 4.10	<i>Rule viewer</i> FIS Sugeno	51
Gambar 4.11	Tampilan Form GUI Sebelum Dijalankan.....	53
Gambar 4.12	Tampilan Form GUI Setelah Dijalankan	54
Gambar 5.1	Variabel Masukan <i>H</i>	61
Gambar 5.2	Variabel <i>Output</i> FIS Mamdani.....	62
Gambar 5.3	Variabel <i>Output</i> FIS Sugeno	63

DAFTAR TABEL

Tabel 2.1 RGB dan <i>range hue</i>	19
Tabel 2.2 Warna kopi Sangat Pekat/ <i>Very Dark</i> , Pekat/ <i>Dark</i> , Biasa/ <i>Middle</i> , Ringan/ <i>Light</i> , dan Sangat Ringan/ <i>Very Light</i> beserta <i>range hue</i> dan <i>intensity</i>	20
Tabel 2.3 Konfigurasi Pin Port pada mikrokontroler ATMEGA 8535	29
Tabel 5.1 Citra kopi yang ditangkap oleh kamera.....	59
Tabel 5.2 <i>Rule</i> yang digunakan pada FIS	64
Tabel 5.3 Hasil Pengujian Kopi untuk Jenis Kopi <i>very dark</i>	65
Tabel 5.4 Hasil Pengujian Kopi untuk Jenis Kopi <i>dark</i>	66
Tabel 5.5 Hasil Pengujian Kopi untuk Jenis Kopi <i>middle</i>	68
Tabel 5.6 Hasil Pengujian Kopi untuk Jenis Kopi <i>light</i>	69
Tabel 5.7 Hasil Pengujian Kopi untuk Jenis Kopi <i>very light</i>	71
Tabel 5.8 Performansi total mesin pengaduk kopi otomatis dengan FIS Mamdani.	73
Tabel 5.9 Pengujian Kopi untuk Jenis Kopi <i>very dark</i>	74
Tabel 5.10 Pengujian Kopi untuk Jenis Kopi <i>dark</i>	75
Tabel 5.11 Pengujian Kopi untuk Jenis Kopi <i>middle</i>	76
Tabel 5.12 Pengujian Kopi untuk Jenis Kopi <i>light</i>	77
Tabel 5.13 Pengujian Kopi untuk Jenis Kopi <i>very light</i>	79
Tabel 5.14 Performansi total mesin pengaduk kopi otomatis dengan FIS Sugeno.....	80

DAFTAR ISTILAH DAN SINGKATAN

<i>Computer vision</i>	Bidang ilmu komputer komputer yang membahas perancangan mesin yang dapat melihat
Besaran	Peristiwa atau kejadian-kejadian yang dapat diukur dan dinyatakan dengan angka
CW	<i>Clock Wise</i> (Searah jarum jam)
CCW	<i>Counter Clock Wise</i> (Berlawanan arah jarum jam)
Citra	Gambar pada bidang 2 dimensi atau fungsi kontinu dari intensitas cahaya pada bidang 2 dimensi
data	Nilai yang mempresentasikan deskripsi suatu objek atau kejadian
Defuzzifikasi	Proses penalaran <i>crisp</i> dimana <i>output</i> yang dihasilkan merupakan suatu nilai tertentu pada <i>domain</i> himpunan <i>fuzzy</i> .
<i>frame</i>	satuan terkecil dalam video.
<i>Frame rate</i>	Jumlah <i>frame</i> yang dilihat setiap detik
<i>Fuzzy Inference System</i>	Sistem yang dapat melakukan penalaran dengan nalurnya berdasarkan kaidah-kaidah linguistik dan memiliki algoritma fuzzy yang menyediakan sebuah aproksimasi untuk dimasuki analisa matematik.
Fuzzifikasi	Memetakan angka-angka masukan <i>crisp</i> ke dalam himpunan Logika <i>Fuzzy</i> .
RGB	<i>Red Green Blue</i>

resolusi

Ukuran *frame Video* (panjang x lebar video)



DAFTAR LAMPIRAN

Lampiran A Program pada GUI Pengaduk Kopi Otomatis

Lampiran B Program coffee.fis

Lampiran C Program coffeesugeno.fis

Lampiran D *Flowchart* Mikrokontroler

Lampiran E Gambar Alat pengaduk kopi

Lampiran F Gambar Rangkaian Sistem



ABSTRACT

The rapid development of technology is linked to the complex human need, one of the examples is the implementation of image processing technology to run an application. In this final submission, an image processing system is developed to run a color-based automatic coffee mixer. First, a user chooses the wanted coffee type by clicking one of the buttons provided in GUI on computer screen, and then the computer will send data to the interface system. Based on the data, interface system will run the dc motor and stepper motor. A camera will capture the image of the coffee that is being mixed in a glass. From the coffee image, the input for the FIS system will be obtained. FIS system will determine the coffee type in the glass. If the coffee in the glass matches the coffee type that the user chose in the beginning, the computer will send data to the interface system and the interface system will stop the rotation of dc motor and stepper motor. If the coffee in the glass has not match the coffee type that the user chose, computer will not send any data to the interface system and will order the camera to capture another image of the coffee. To control the coffee mixer, ATMEGA8535 is used. The coffee mixer works automatically according to the input of the computer. The decision maker system is using Fuzzy Inference System FIS). This system has been tested 20 times for each coffee type with success percentage 92% for FIS Mamdani and 93% for FIS Sugeno.

Keyword : color recognition, FIS Mamdani, FIS Sugeno, image processing application, microcontroller.

ABSTRAK

Perkembangan teknologi yang semakin pesat berkaitan dengan semakin kompleksnya kebutuhan manusia, salah satunya adalah penerapan teknologi pengolahan citra dalam menjalankan suatu aplikasi. Pada tugas akhir ini, dikembangkan sistem pengolahan citra untuk menjalankan alat pengaduk kopi otomatis berdasarkan warna. Pertama-tama user akan memilih jenis kopi yang diinginkan dengan mengklik salah satu tombol pada GUI kemudian komputer akan mengirimkan data ke antarmuka. Berdasarkan data tersebut, antarmuka akan menggerakkan motor dc dan stepper. Kamera akan meng-capture gambar kopi yang sedang diaduk. Dari citra kopi tersebut diperoleh input untuk sistem FIS. FIS akan memutuskan jenis kopi dalam gelas. Apabila kopi telah cocok dengan pilihan user, komputer akan mengirimkan data ke antarmuka dan antarmuka akan menghentikan motor dc dan stepper. Apabila kopi belum cocok dengan pilihan user, program akan kembali memerintahkan kamera untuk meng-capture kembali gambar kopi. Pengendalian alat pengaduk kopi otomatis menggunakan mikrokontroler ATMEGA8535. Alat pengaduk kopi otomatis bekerja berdasarkan input dari komputer. Sistem pengambilan keputusan pada sistem menggunakan Fuzzy Inference System (FIS). Sistem ini sudah diujikan sebanyak 20 kali untuk setiap jenis kopi yang diujikan dengan persentase keberhasilan 92% untuk FIS Mamdani, dan 93% untuk FIS Sugeno.

Kata kunci: Pengenalan warna, FIS Mamdani, FIS Sugeno, aplikasi pengolahan citra, mikrokontroler.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kemajuan teknologi pengenalan memberikan banyak kontribusi akan kemudahan pengenalan informasi pada suatu objek oleh sebuah mesin. Saat ini, dalam melakukan kegiatan sehari-hari baik itu kegiatan pribadi maupun kegiatan yang berhubungan dengan pekerjaan, manusia telah banyak dibantu oleh hasil dari kemajuan teknologi dan informasi itu sendiri. Contohnya dalam beberapa aktivitas manusia hanya perlu memencet beberapa tombol atau hanya dengan bersuara maka apa yang ia inginkan dilakukan oleh alat-alat yang menjadi pengganti tangan, kaki atau pikiran.

Salah satu contoh kemajuan teknologi pengenalan adalah sistem visual mesin (*machine vision*). Visual mesin, merupakan salah satu fasilitas yang diberikan pada mesin untuk dapat mengenali area disekitarnya layaknya visual manusia, dan mempunyai persepsi yang sama dengan mata manusia dalam mengamati sebuah objek^[2]. Contoh lainnya adalah teknologi *computer vision*. *Computer vision* merupakan istilah dalam proses yang bertujuan untuk membuat keputusan yang berguna mengenai fisik sebenarnya dari sebuah objek dan tentang pemandangan berdasarkan *image* yang didapat dari sensor seperti kamera atau yang lainnya. Pada hakikatnya, *computer vision* mencoba meniru cara kerja sistem visual manusia (*human vision*). --

Telah banyak penelitian terdahulu yang memanfaatkan teknologi pengenalan, di antaranya adalah Alexandre R.J. Francois and Gerard G. Medioni^[3] dalam

jurnalnya yang berjudul *“Adaptive Color Background Modeling Real-Time Segmentation of Video Stream”* yang membahas *segmentation and real-time background modeling* yang bekerja pada ruang warna HSV (*Hue-Saturation-Value*) dan RGB (*Red-Green-Blue*) dan Aditya^[1], dalam penelitiannya yang berjudul *“Sistem Kamera Penjejak Objek berdasarkan warna menggunakan motor DC”* membahas mengenai *tracking* sebuah objek oleh suatu robot mobil berdasarkan pada identifikasi warna.

Logika Fuzzy adalah metodologi sistem kontrol pemecahan masalah. Dalam logika klasik dinyatakan bahwa segala sesuatu bersifat biner (bernilai 0 atau 1). Namun dalam logika Fuzzy memungkinkan nilai keanggotaan berada di antara 0 dan 1. Logika Fuzzy merupakan suatu sistem yang bisa memberikan suatu keputusan yang baik untuk kondisi yang tidak pasti (tidak tentu), mudah dimengerti, memiliki toleransi terhadap data-data yang tidak tepat sehingga banyak digunakan pada penelitian di berbagai bidang disiplin ilmu, di antaranya pada pemrosesan citra^[14]. Logika Fuzzy telah banyak digunakan pada penelitian-penelitian terdahulu, di antaranya Dwi Purnomo^[18] dalam jurnalnya yang berjudul *“ Sistem Pakar Fuzzy Penentuan dan Peningkatan Kualitas Manggis”* membahas mengenai aplikasi logika fuzzy dan pengolahan citra untuk menentukan dan meningkatkan kualitas buah manggis dan Lestari, Dwi Rili^[15] dalam penelitiannya yang berjudul *“Perencanaan Sistem Pengenalan Penyakit Darah”* membahas mengenai aplikasi logika Fuzzy dan pengolahan citra pada citra darah mikroskopis *digital*.

Berdasarkan penelitian-penelitian tersebut, penulis tertarik untuk membuat tugas akhir dengan mengaplikasikan pemrosesan citra dan logika Fuzzy pada

mesin pengaduk kopi otomatis. Pada pengaduk kopi ini, input berasal dari komputer yang dihubungkan dengan mesin pengaduk. Pengaduk akan bergerak sesuai input yang diberikan oleh manusia melalui komputer, dan akan berhenti mengaduk secara otomatis berdasarkan tingkat kecoklatan warna kopi yang ditangkap oleh kamera. Proses pengenalan warna kopi menggunakan logika Fuzzy.

I.2 Tujuan Penelitian

Tujuan akhir dari penelitian ini adalah membuat mesin pengaduk kopi yang menggunakan pemrosesan citra sebagai kontrol pergerakan pengaduk kopi.

I.3 Manfaat penelitian

Manfaat penelitian ini adalah:

1. Dapat dijadikan sebagai salah satu referensi untuk aplikasi pemrosesan citra pada mesin atau robot sederhana.
2. Dapat dijadikan referensi untuk pengembangan mesin pengaduk kopi otomatis yang fleksibel terhadap ukuran gelas, volume air, dan kepekatan kopi.

I.4 Batasan Masalah

Untuk mengarahkan pokok permasalahan dalam penelitian ini, penulis mengambil batasan masalah sebagai berikut :

1. Jenis kopi yang digunakan dalam penelitian ini adalah Sangat Pekat/ *Very Dark*, Pekat/*Dark*, Biasa/*Middle*, dan Ringan/ *Light*, dan Sangat Ringan/*Very Light*.
2. Kondisi awal kopi adalah pada jenis Sangat Pekat/*Very Dark*.
3. Gelas yang digunakan adalah gelas bening.

4. Jenis kamera yang digunakan adalah kamera tunggal jenis *web camera* yang dengan resolusi video adalah 160x120 piksel, dengan *frame rate* 30 fps.
5. Metoda penentuan jenis kopi berdasarkan Logika Fuzzy dengan ruang warna HIS.
6. Mikrokontroler yang dipakai adalah jenis ATMEGA8535.
7. *Software* yang dipakai dalam pemrograman ini adalah Matlab 7.0.1 dan bahasa c.

1.5 Metodologi Penelitian

Metodologi penelitian yang digunakan pada tugas akhir ini adalah:

1. Studi literatur sebagai gambaran awal tentang pemahaman konsep dari masalah yang akan dibahas pada tugas akhir ini
2. Pembuatan simulasi alat
3. Pembuatan program
4. Pengujian simulasi alat
5. Pembuatan alat
6. Pengujian alat
7. Penyusunan laporan akhir.

1.6 Sistematika Penulisan

Tugas akhir ini disusun dengan sistematika sebagai berikut :

1. Bab I Pendahuluan

Bab ini berisi latar belakang penelitian, perumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, dan sistematika penulisan.

2. Bab II Tinjauan Pustaka

Bab ini berisi teori dasar yang mendukung penelitian tugas akhir ini.

3. Bab III Metodologi Penelitian

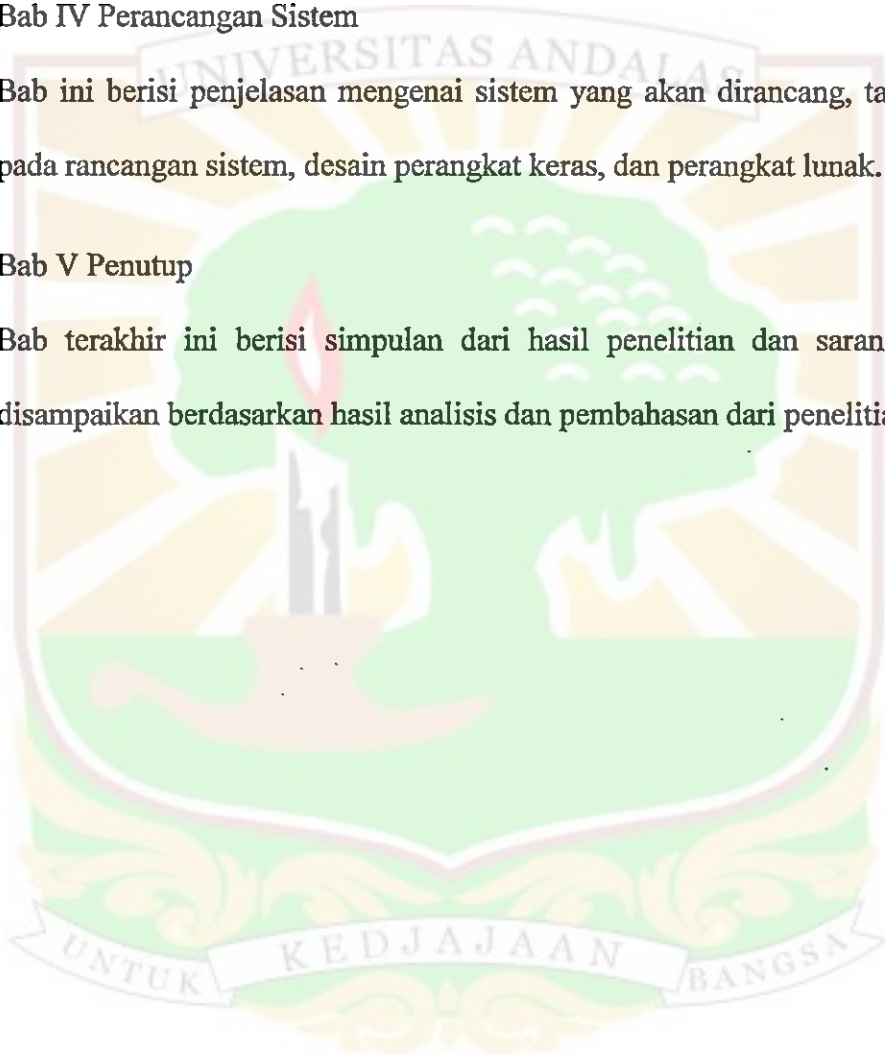
Bab ini berisi metodologi yang digunakan dalam penelitian tugas akhir.

4. Bab IV Perancangan Sistem

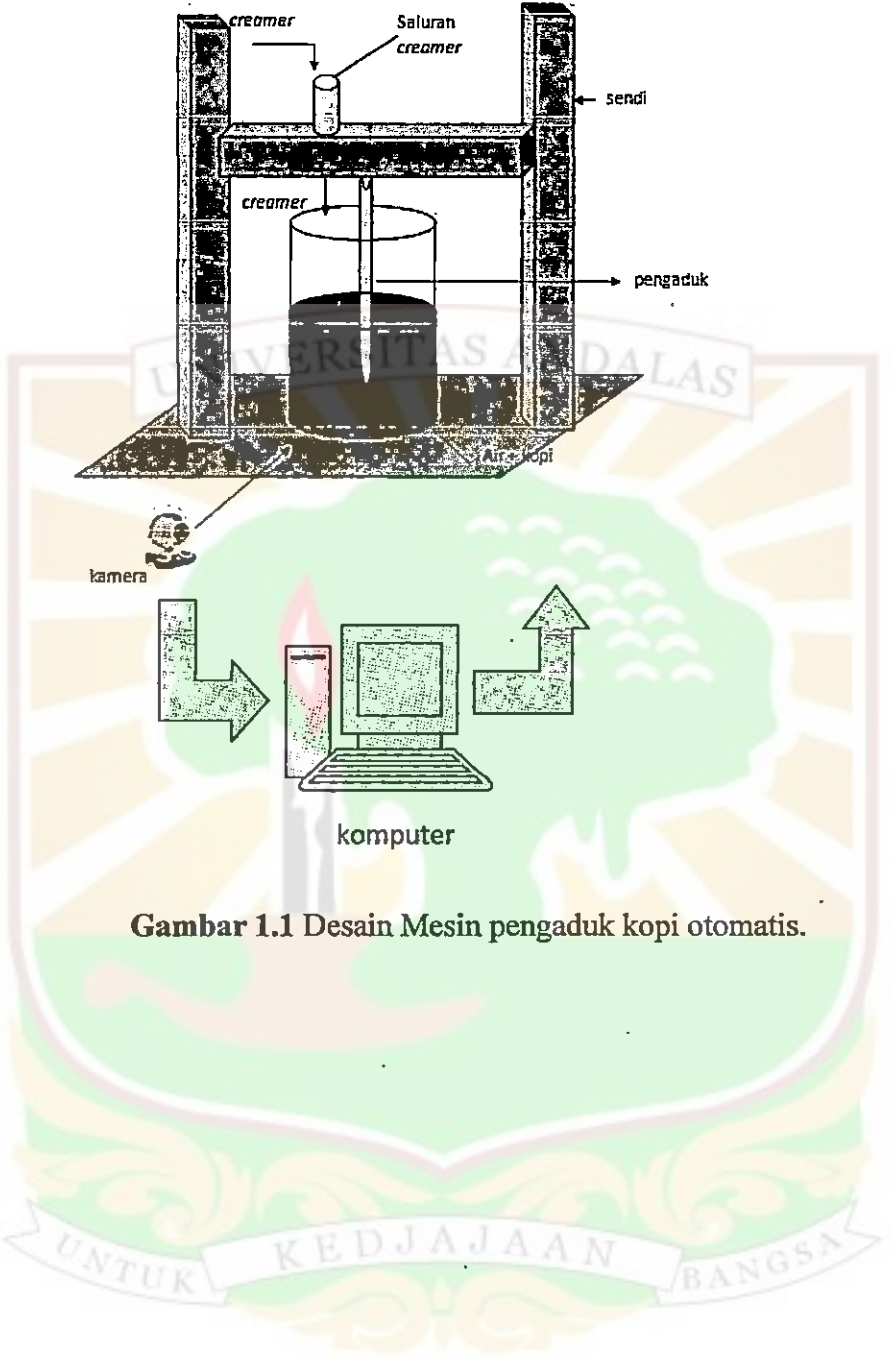
Bab ini berisi penjelasan mengenai sistem yang akan dirancang, tahapan pada rancangan sistem, desain perangkat keras, dan perangkat lunak.

5. Bab V Penutup

Bab terakhir ini berisi simpulan dari hasil penelitian dan saran yang disampaikan berdasarkan hasil analisis dan pembahasan dari penelitian ini.



1.7 Desain Mesin



Gambar 1.1 Desain Mesin pengaduk kopi otomatis.

BAB II

TINJAUAN PUSTAKA

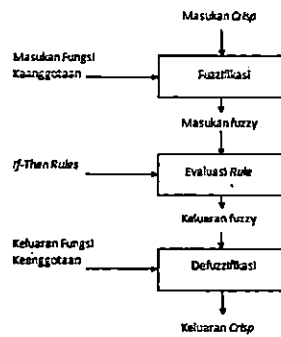
2.1 Logika Fuzzy

Logika Fuzzy pertama kali dikembangkan tahun 1965 oleh Prof. Lotfi Zadeh dari University of California. Ide dari logika fuzzy adalah bahwa di alam ini tidak ada sesuatu yang mutlak '*benar*' atau mutlak '*salah*'. Antara '*benar*' dan '*salah*' ada suatu gradasi (perubahan) sehingga suatu kejadian bisa dinyatakan sebagai sebagian '*benar*' dan sebagian '*salah*' secara bersamaan. Logika fuzzy merupakan suatu konsep dari "ketidakpastian"^[5].

2.2 Penerapan Logika Fuzzy

Metode fuzzy menggunakan pernyataan-pernyataan jika-maka (*if-then*) untuk menyatakan hubungan antara masukan dengan keluaran sistem. Pernyataan-pernyataan ini dibangun berdasarkan pengalaman, pengetahuan ataupun intuisi (*common sense*) dari seorang pakar. Pengetahuan yang didapat bisa berasal dari percobaan-percobaan yang dilakukan untuk mengenali tingkah laku (*behaviour*) dari proses.

Motivasi utama teori fuzzy adalah memetakan sebuah ruang *input* ke dalam ruang *output* dengan menggunakan *if-then rules*. Pemetaan dilakukan dalam suatu *Fuzzy Inference System* (FIS). FIS mengevaluasi semua *rule* secara simultan untuk menghasilkan kesimpulan.^[5] Konsep logika Fuzzy ditampilkan pada gambar 2.1

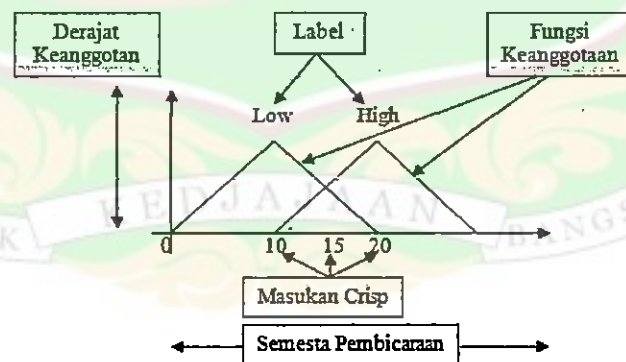


Gambar 2.1 Konsep dari Logika Fuzzy

Pernyataan-pernyataan *'if-then'* pada logika fuzzy disebut *rules*. Pernyataan *'if'* digunakan untuk menggambarkan kondisi sistem yang terjadi. Pernyataan *'then'* digunakan untuk menggunakan aksi yang dilakukan oleh sistem.

Logika fuzzy menyatakan perubahan secara gradual sehingga setiap kondisi mempunyai derajat yang disebut sebagai Derajat Keanggotaan (*Degree of membership*). Fungsi yang menyatakan hubungan antara suatu kondisi dengan derajatnya disebut sebagai Fungsi Keanggotaan (*Membership Function*). Fungsi Keanggotaan ini bisa berbentuk segitiga, trapesium, *singleton*, maupun kurva.

Bagian-bagian dari Fungsi Keanggotaan digambarkan pada gambar 2.2.



Gambar 2.2 Bagian-Bagian Fungsi Keanggotaan^[18]

Logika *fuzzy* tidak hanya mengenali batasan yang jelas dari Fungsi Keanggotaan sebagai alternatif dari keanggotaan hitam atau putih, tapi juga gradasi tanpa batas yang ada diantaranya. Hal ini tampaknya tidak jelas, tapi

logika fuzzy mengurangnya dengan memberikan nilai yang spesifik dari setiap gradasi tersebut. Untuk lebih memahami sebuah sistem *fuzzy*, maka diperlukan konsep dasar yang berhubungan dengan logika *fuzzy* tersebut.

1. Label. Merupakan deskripsi dari nama yang digunakan untuk mengidentifikasi *membership function*. Label-label ini menunjukkan suatu keadaan misalnya *very low, low, medium, high, very high*.

2. Derajat Keanggotaan (*Degree of Membership*). Menyatakan derajat keanggotaan dari *crisp input* yang sesuai dengan *membership function* dengan nilai antara 0 sampai dengan 1. Juga disebut sebagai *membersip grade, truth value*, atau *fuzzy input*.

3. Masukan *Crisp* (*Crisp input*). Merupakan nilai masukan dari proses fuzzy yang merupakan besaran suatu kondisi nyata.

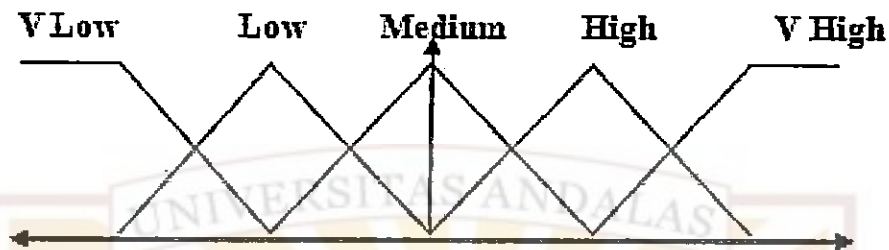
4. Fungsi Keanggotaan (*Membership Function*). Mendefenisikan *fuzzy set* dengan memetakan *crisp input* dari nilai domainnya kedalam derajat keanggotaan (*degree of membership*).

5. Semesta Pembicaraan (*Universal of Discourse*). Adalah *range* dari semua nilai yang mungkin dipakai dalam variabel sistem, merupakan semesta dari himpunan *input*.

2.2.1 Fuzzifikasi

Proses fuzzifikasi adalah proses pengubahan masukan *crisp* menjadi masukan *fuzzy*. Untuk mentransformasikan masukan *crisp* menjadi masukan *fuzzy*, diperlukan Fungsi Keanggotaan untuk setiap masukan. Proses fuzzifikasi mengambil nilai masukan *crisp* dan membandingkannya dengan Fungsi Keanggotaan yang telah ada untuk menghasilkan harga masukan *fuzzy*.

Dalam proses fuzzifikasi dapat diolah lebih dari satu macam masukan *crisp*. Fungsi Keanggotaan dari masukan *fuzzy* dapat digambarkan oleh gambar 2.3.



Gambar 2.3 Fungsi Keanggotaan

2.2.2 Evaluasi Rule (Rule Evaluation)

Pada evaluasi *rule* ini, *fuzzy* menggunakan *rules* yang telah dibuat untuk menentukan aksi yang harus dilakukan. Pada tahap ini dilakukan evaluasi tiap *rule* dengan masukan yang dihasilkan dari proses fuzzifikasi.

Proses selanjutnya adalah menentukan keluaran *fuzzy* dengan cara membandingkan *rule strength* dari *rule* yang mempunyai label *consequent* sama. Proses Fuzzifikasi dapat dilihat pada gambar 2.4.



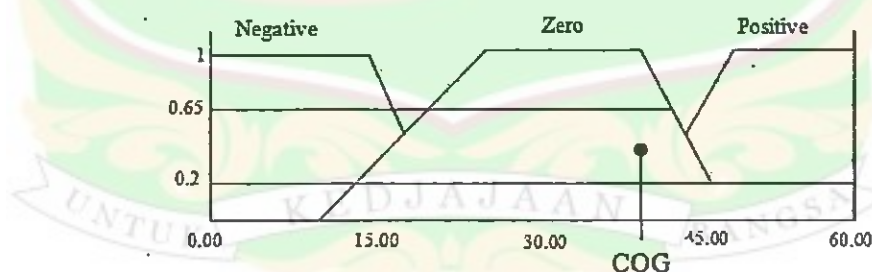
Gambar 2.4 Flow Diagram Proses Fuzzifikasi^[18]

2.2.3 Defuzzifikasi

Pada proses defuzzifikasi, semua nilai keluaran *fuzzy* yang dihasilkan proses evaluasi *rule* dikombinasikan dengan kealuaran Fungsi Keanggotaan untuk mendapatkan keluaran sesuai sistem yang diinginkan. Ada dua metode dalam proses defuzzifikasi yaitu :

1. **Metode Maximizer.** Metode ini mengambil lokasi dari nilai tertinggi (*maximum value*) pada *fuzzy output* sebagai hasil akhirnya.

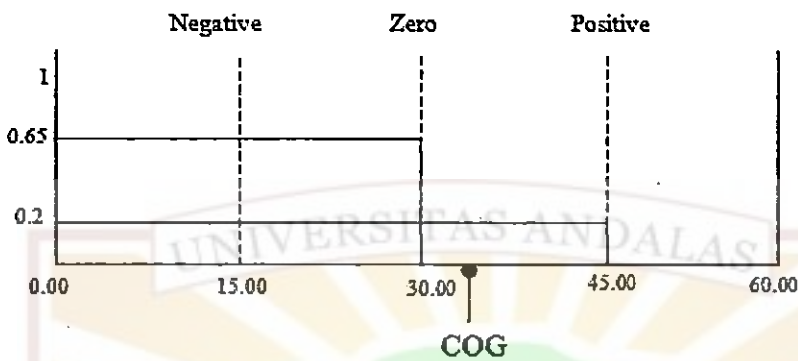
2. **Metode Centroid (Center of Gravity).** Pada metode ini setiap fungsi keanggotaan *output* yang mempunyai nilai di atas nilai *fuzzy output* akan dipotong. Pemotongan ini disebut *lambda cut*. Hasilnya (fungsi keanggotaan yang telah terpotong) digabungkan lalu baru dihitung *Center of Gravity* keseluruhannya. Di bawah ini adalah gambar dari pemotongan *output* fungsi keanggotaan (*lambda cut*), penggabungan keanggotaan yang telah dipotong dan titik berat COG dari daerah yang ditunjuk. Metode *Center of Gravity* ditunjukkan pada gambar 2.5.



Gambar 2.5 Metode *Center of Gravity*

Metode defuzzifikasi *Center of Gravity* (COG) dapat juga diterapkan pada *output* fungsi keanggotaan yang berbentuk *singleton*. *Output* fungsi keanggotaan yang berbentuk *singleton* merupakan sebuah garis vertikal tunggal yang tidak mempunyai bobot. Pemotongan (*lambda cut*) pada *output* fungsi keanggotaan

dilakukan dengan pengurangan tinggi dari garis vertikal tersebut, seperti terlihat pada gambar 2.6 di bawah ini.



Gambar 2.6 Metoda *Center of Gravity* pada *Singleton*

Pada *output* fungsi keanggotaan *Singleton* ini untuk mendapatkan harga *crisp output* cukup dengan menjumlahkan hasil perkalian dari *fuzzy output* dengan koordinatnya dari posisi *singleton*, kemudian hasilnya dibagi dengan *fuzzy output*. Untuk Gambar 2.6 digunakan persamaan *Weighted Average* sebagai berikut:

$$Crisp\ Output = \frac{\sum_{i=1}^N w_i \times z_i}{\sum_{i=1}^N z_i} \dots\dots\dots (2.1)$$

Untuk Gambar 2.6 di atas hasilnya adalah :

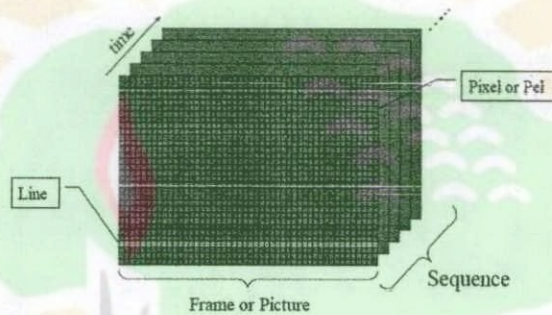
$$Crisp\ Output = \frac{0 \times 0 + 0.65 \times 30 + 0.2 \times 45}{0 + 0.65 + 0.2}$$

2.3 Video

2.3.1 Pengertian Video

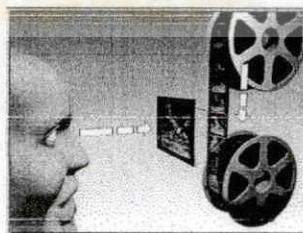
Video merupakan serangkaian citra diam (*frame*) yang ditampilkan secara beruntun dalam suatu interval waktu dengan kecepatan tertentu, sehingga memberi kesan pada mata kita sebagai gambar yang bergerak (citra bergerak).

Video adalah teknologi pemrosesan sinyal elektronik yang mewakilkan citra bergerak. Video terdiri atas piksel-piksel yang tersusun menjadi *array* tiga dimensi. Dua dimensi menunjukkan gambar spasial (horizontal dan vertikal) dan satu dimensi lainnya merepresentasikan waktu atau jumlah gambar. Untuk menganalisa suatu video, diperlukan pembagian video menjadi unit yang lebih kecil dan memiliki beberapa hubungan semantik seperti *shot*, *key frame* dan *scene*. Komponen video ditunjukkan oleh gambar 2.7.



Gambar 2.7 Komponen Video^[2]

Key frame adalah suatu *frame* yang menunjukkan bagian penting dari *shot*. Dalam satu *shot* dapat terdiri dari beberapa *key frame* tergantung kompleksitas *shot*. Beberapa *shot* dipisahkan oleh *shot boundary*. Sedangkan video *scene* adalah koleksi dari *shot* berdekatan yang memiliki hubungan semantik. Gambar 2.8 mengilustrasikan bagaimana proses *single frame* yang dirangkai menjadi video.



Gambar 2.8 *Single Frame* yang dirangkai menjadi Video

2.3.2 Pengambilan *Frame* pada Video

Video digital diambil dengan menggunakan alat penangkap video (video imager) yaitu *camcorder* seperti mini DV, kamera digital, DVD *camcorder*, *handycam*, *web camera* dan lainnya. Proses yang terjadi adalah ketika kamera ditujukan pada objek yang akan diambil maka benda tersebut akan memantulkan cahaya ke kamera sebesar intensitas cahaya yang diterima benda dikalikan dengan cahaya yang dipantulkan benda terhadap kamera.

Yao Wang, dkk. dalam bukunya “*Video Processing and Communications*” menyebutkan bahwa cahaya yang diterima benda dari sumber cahaya adalah sebesar $E(x, y, z, \lambda)$, dimana: (x,y,z) adalah titik koordinat dan λ adalah panjang gelombang sumber. Sedangkan cahaya yang dipantulkan benda adalah $r(x,y,z,\lambda)$. Persamaan 2.1 menerangkan nilai sinar yang ditangkap oleh video kamera. Gambar 2.9 menggambarkan proses penangkapan *image* objek pada kamera.

$$c(x, y, z, \lambda) = E(x, y, z, \lambda) \times r(x, y, z, \lambda) \dots \dots \dots (2.1)$$



$E(x,y,z,\lambda)$
 $c(x,y,z,\lambda) = E(x,y,z,\lambda) \cdot r(x,y,z,\lambda)$

Gambar 2.9 Proses Penangkapan *Image* Objek Pada Kamera^[2]

2.3.3 Pengolahan Citra dengan Logika Fuzzy

Format warna pada *frame* adalah RGB (*Red, Green, Blue*) sehingga sebelum diproses dengan logika Fuzzy kita harus mengkonversi format warna RGB ke dalam format HLS (*Hue, Lightness, Saturation*). Persamaan 2.2, 2.3, dan 2.4 berikut formula untuk mengubah format RGB ke HIS^[13].

$$H = \cos^{-1} \left(\frac{0.5((R - G) + (R - B))}{\sqrt{(R - G)^2 + (R - B)(G - B)}} \right) \dots\dots\dots (2.2)$$

$$I = \frac{R + G + B}{3} \dots\dots\dots (2.3)$$

$$S = 1 - 3 \frac{\min\{R, G, B\}}{R + G + B} \dots\dots\dots (2.4)$$

Nilai HIS terdiri atas nilai *hue*, yang berhubungan dengan warna objek, nilai intensitas yang menunjukkan seberapa suatu warna mengandung warna putih dan nilai saturasi yang berhubungan dengan kejernihan warna dalam cahaya ^[12]. Nilai *hue*, *intensity* dan *saturation* ini disebut dengan karakteristik citra.

Setelah didapatkan masukan citra dan pembentukan karakteristik citra, hasil data-data tersebut dikumpulkan menjadi suatu database yang kemudian digunakan untuk membentuk suatu fungsi keanggotaan pada *Fuzzy Inference System* (FIS) yang akan dibuat.

FIS adalah sebuah metode yang merepresentasikan harga-harga dalam bentuk vektor *input*, menarik kesimpulan berdasarkan sekumpulan *if-then rules* dan kemudian menghasilkan vektor *output*^[18].

2.3.4 Histogram citra

Histogram citra adalah grafik yang menggambarkan penyebaran nilai-nilai intensitas piksel dari suatu citra atau bagian tertentu di dalam citra. Citra dibentuk dari oleh piksel-piksel yang merupakan bagaian terkecil dari sebuah citra, jadi pengolahan citra dapat dikatakan sebagai pengolahan piksel-piksel. Untuk dapat mengolah piksel inilah, informasi tentang sebaran piksel dengan intensitas tertentu dalam suatu citra harus diketahui terlebih dahulu. Informasi ini dapat diperoleh

dengan tepat dari sebuah histogram. Histogram citra akan menampilkan frekuensi kemunculan relatif dari intensitas pada citra tersebut. Selain itu histogram juga mampu menampilkan tingkat kecerahan dan kontras citra.

2.3.5 Sistem Komponen ruang warna

Model warna yang banyak digunakan saat ini umumnya berorientasi pada hardware (contoh monitor dan printer) atau aplikasi dimana manipulasi warna menjadi tujuannya (kreasi warna grafik untuk animasi). Berikut merupakan model-model warna.

Model Warna berorientasi *hardware* :

1. Model RGB (*Red, Green, Blue*) untuk warna monitor dan warna pada kamera video.
2. Model CMY (*Cyan, Magenta, Yellow*) untuk model printer.
3. Model YIQ, digunakan untuk standard televisi. Y berkoresponden dengan luminasi, I dan Q adalah dua komponen kromatik yang disebut *Inphase* dan *Quarature*.

Model warna berorientasi *Software* (*hue, saturation, brightness*) adalah manipulasi Model HSV (*Hue, Saturation, Value*) / HSI (*Hue, Saturation, Intensity*) / HLS (*Hue, Lightness, Saturation*). Berikut standar komponen warna yang digunakan dalam penelitian ini:

2.3.5.1 RGB (*red, green, blue*)

Citra berwarna dikenal dengan nama citra spektral, karena warna pada citra disusun oleh tiga komponen warna yang disebut komponen RGB, yaitu merah (*red*), hijau (*green*), dan biru (*blue*). Pada citra ini masing – masing warna primernya yaitu merah, hijau, dan biru terdiri dari 8 bit dan kemudian warna –

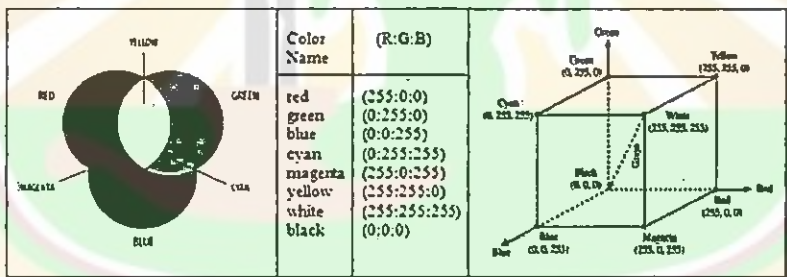
warna ini digabungkan untuk menghasilkan warna yang sesuai. Oleh karena itu, citra RGB terdiri dari 24 bit per piksel (bpp).

Sebuah jenis warna, dapat kita bayangkan sebagai sebuah vektor di ruang dimensi-3. Yang biasanya dipakai dalam matematika, koordinat-ujungnya dinyatakan dalam bentuk tiga bilangan, yaitu komponen-x, komponen-y dan komponen-z. Misalnya, sebuah vektor dituliskan sebagai $r = (x,y,z)$. Untuk warna, komponen-komponen tersebut digantikan oleh komponen R(Red), G(Green), B(Blue).

Jadi, sebuah jenis warna ditulis misalnya sebagai:

$$\text{warna} = \text{RGB}(30, 75, 255).$$

Dalam ungkapan ini warna putih = $\text{RGB}(255,255,255)$, sedangkan warna hitam = $\text{RGB}(0,0,0)$. Gambar 2.10 menggambarkan ruang warna RGB.



Gambar 2.10 Ruang warna RGB

Dasar pengolahan citra adalah pengolahan warna RGB (*Red, Green, Blue*) pada posisi tertentu. Dalam pengolahan citra, warna direpresentasikan dengan nilai *hexadesimal* dari 0x00000000 sampai 0x00FFFFFF. Warna hitam adalah 0x00000000 dan warna putih adalah 0x00FFFFFF. Berikut terlihat definisi nilai warna dan variabel 0x00 menyatakan angka dibelakangnya adalah *hexadecimal*, yang juga merupakan aturan penulisan kombinasi warna RGB.

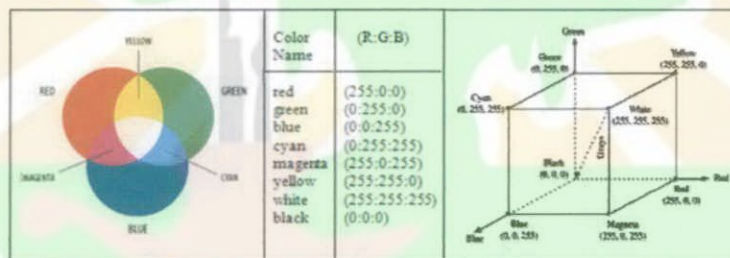
warna ini digabungkan untuk menghasilkan warna yang sesuai. Oleh karena itu, citra RGB terdiri dari 24 bit per piksel (bpp).

Sebuah jenis warna, dapat kita bayangkan sebagai sebuah vektor di ruang dimensi-3. Yang biasanya dipakai dalam matematika, koordinat-ujungnya dinyatakan dalam bentuk tiga bilangan, yaitu komponen-x, komponen-y dan komponen-z. Misalnya, sebuah vektor dituliskan sebagai $r = (x,y,z)$. Untuk warna, komponen-komponen tersebut digantikan oleh komponen R(Red), G(Green), B(Blue).

Jadi, sebuah jenis warna ditulis misalnya sebagai:

warna = RGB(30, 75, 255).

Dalam ungkapan ini warna putih = RGB(255,255,255), sedangkan warna **hitam** = RGB(0,0,0). Gambar 2.10 menggambarkan ruang warna RGB.



Gambar 2.10 Ruang warna RGB

Dasar pengolahan citra adalah pengolahan warna RGB (*Red, Green, Blue*) pada posisi tertentu. Dalam pengolahan citra, warna direpresentasikan dengan nilai *hexadesimal* dari 0x00000000 sampai 0x00FFFFFF. Warna hitam adalah 0x00000000 dan warna putih adalah 0x00FFFFFF. Berikut terlihat definisi nilai warna dan variabel 0x00 menyatakan angka dibelakangnya adalah *hexadecimal*, yang juga merupakan aturan penulisan kombinasi warna RGB.

0 x 00

XX

XX

XX

Nilai *Blue*

Nilai *Green*

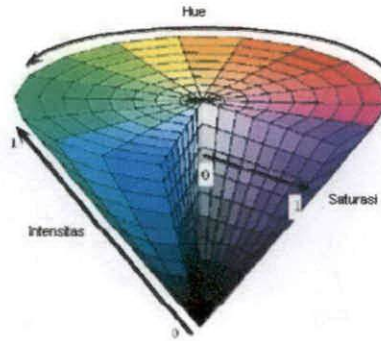
Nilai *Red*

Setiap warna mempunyai *range* nilai 00 (angka desimalnya adalah 0) dan FF (angka desimalnya adalah 255) atau mempunyai nilai derajat keabuan (*gray scale*) $256 = 2^8$. Dengan demikian, *range* warna yang digunakan adalah $2^8 \cdot 2^8 \cdot 2^8 = 2^{24}$ (atau dikenal dengan istilah *true color* pada *windows*). Warna suatu target ditentukan dengan menentukan nilai masing-masing komponen RGB-nya. Misalnya kita ingin menyeleksi warna biru. RGB dari warna biru yaitu R = 0-50, G = 0-50, dan B = 200-250. Nilai-nilai RGB ini disebut dengan *thresholding*.

2.3.5.2 HSI (*Hue Saturation Intensity*) / HSV / HLS

Sistem HSI adalah cara lain penggunaan ruang warna pada pengolahan citra, yang lebih intuitif untuk visi manusia. Sistem warna HSI merupakan warna pelengkap yang dihasilkan dari pengaturan warna RGB yang diproyeksikan ke diagonal utama yaitu dari warna hitam ke warna putih.

Sistem HSI memisahkan informasi warna sebuah citra dari informasi intensitasnya. Informasi warna diwakili dengan nilai *hue* dan saturasi, sedangkan intensitas, mencitrakan kecerahan sebuah citra, yang ditentukan dengan kuantitas cahaya. Model warna HSI lebih berorientasi *software* sehingga lebih mudah untuk dimanipulasi. Dalam kondisi tertentu, salah satu komponen HSI akan saling meniadakan. Jika suatu warna memiliki nilai saturasi nol, maka otomatis warna tersebut tidak memiliki nilai *hue*. Ruang warna HIS digambarkan oleh gambar 2.11.



Gambar 2.11 HSI color space

Hue menyatakan spektrum warna dominan dalam cahaya. *Hue* menunjukkan kandungan warna sebenarnya dari suatu citra. Nilai warna dalam *hue* dari 0 derajat sampai 360 derajat. Misalkan warna merah bernilai 0 derajat pada *hue*. *Saturation* menyatakan kejernihan spektrum warna dalam cahaya (ukuran kemurnian suatu warna). Semakin jernih suatu warna, maka nilai saturasinya semakin tinggi. Tetapi bila suatu warna tingkat kejernihannya kurang, maka nilai saturasinya kurang dari 255 sampai 0 (warna hitam). *Intensity* atau *value* menyatakan tingkat kecerahan cahaya dari warna yang dilihat dari batas yang paling redup atau gelap sampai batas yang paling terang

Tabel 2.1 menerangkan *range* ruang warna RGB dan *Hue*.

Tabel 2.1 RGB dan *range hue*

Warna	RGB	<i>Hue</i>
	0,0,0	
	255,0,0	0 – 5
	0,255,0	80 – 110
	0,0,255	120 – 160
	128,0,128	170 – 210
	255,255,0	20 – 60
	150,75,0	10 – 20
	255,255,255	-

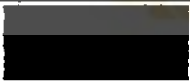
2.3.5.3 Kopi Pekat/*Black*, Biasa/*Medium*, dan Ringan/ *Light*.

Kopi merupakan salah satu jenis polong-polongan dengan kandungan senyawa kompleks di antaranya kafein dan *chlorogenic acid*. Krimer adalah salah satu bahan campuran yang sering digunakan untuk menambah cita rasa kopi. Komponen penyusun utamanya terdiri dari tepung sirup jagung, minyak nabati dan kaseinat dengan bahan tambahan berupa bahan pengemulsi, anti kempal, dan bahan pewarna^[7]. Pada penelitian ini, kopi hitam biasa dibagi menjadi lima kategori berdasarkan kuantitas krimer yang dimasukkan ke dalamnya, yaitu:

1. Kopi sangat pekat/ *very dark*
2. Kopi pekat/ *dark*
3. Kopi Biasa/*Middle*
4. Kopi Ringan/*Light*
5. Kopi sangat ringan / *very light*.

Tabel 2.2 menunjukkan warna kopi Sangat Pekat/ *Very Dark*, Pekat/*Dark*, Biasa/*Middle*, Ringan/*Light*, dan Sangat Ringan/ *Very Light* beserta *range hue*-nya.

Tabel 2.2 Warna kopi Sangat Pekat/ *Very Dark*, Pekat/*Dark*, Biasa/*Middle*, Ringan/*Light*, dan Sangat Ringan/ *Very Light* beserta *range hue*-nya.

Jenis Kopi	Warna	Hue	Intensity
<i>Very dark</i> /sangat pekat		0.117 - 0.16	46-56
<i>Dark</i> / pekat		0.0933 - 0.1276	50-78

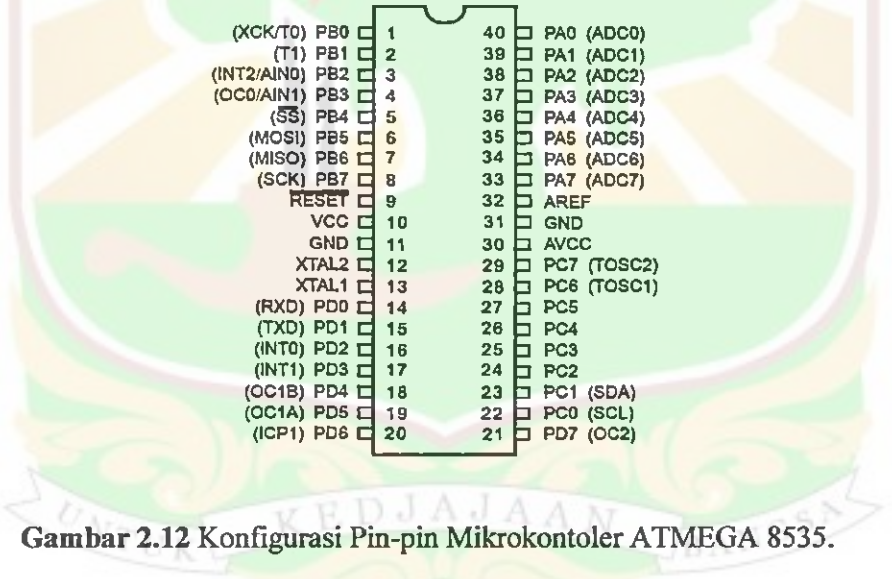
<i>Medium/biasa</i>		0.0794 - 0.104	68-95
<i>Light/Ringan</i>		0.0599 - 0.087	89-105
<i>Very light / sangat ringan</i>		0.0005 - 0.0717	98-120

2.4 Antarmuka

2.4.1 Mikrokontroler ATMEGA 8535

Mikrokontroler ATMEGA 8535 merupakan salah satu mikrokontroler 8 bit yang memiliki ADC internal dan komunikasi serial dengan kecepatan hingga 2.5 Mbps serta diproduksi oleh ATMEL.

2.4.1.1 Konfigurasi Mikrokontroler ATMEGA 8535



Gambar 2.12 Konfigurasi Pin-pin Mikrokontroler ATMEGA 8535.

Pada gambar 2.12 dapat dilihat pin-pin yang terdapat pada mikrokontroler ATMEGA 8535, diantaranya :

- VCC (*power supply*)

➤ GND (*ground*)

➤ Port A (PA7..PA0)

Port A berfungsi sebagai *input* analog pada A/D Konverter. Port A juga berfungsi sebagai suatu *port* I/O 8-bit dua arah, jika A/D Konverter tidak digunakan. Pin - pin *port* dapat menyediakan resistor *internal pull-up* (yang dipilih untuk masing-masing bit). Port A *output buffer* mempunyai karakteristik gerakan simetris dengan keduanya *sink* tinggi dan kemampuan sumber. Ketika pin PA0 ke PA7 digunakan sebagai *input* dan secara *eksternal* ditarik rendah, pin - pin akan memungkinkan arus sumber masuk jika resistor *internal pull-up* diaktifkan. Pin *port* A adalah *tri-stated* jika kondisi *reset* menjadi aktif, sekalipun waktu habis.

➤ Port B (PB7..PB0)

Port B adalah suatu *port* I/O 8-bit dua arah dengan resistor *internal pull-up* (yang dipilih untuk beberapa bit). Port B (*output buffer*) mempunyai karakteristik gerakan simetris dengan keduanya *sink* tinggi dan kemampuan sumber. Pin *port* B adalah *tri-stated* jika kondisi *reset* menjadi aktif, sekalipun waktu habis.

➤ Port C (PC7..PC0)

Port C adalah suatu Port I/O 8-bit dua arah dengan resistor *internal pull-up* (yang dipilih untuk beberapa bit). Port C *output buffer* mempunyai karakteristik gerakan simetris dengan keduanya *sink* tinggi dan kemampuan sumber. Pin Port C juga adalah *tri-stated* jika kondisi *reset* menjadi aktif, sekalipun waktu habis.

➤ Port D (PD7..PD0)

Port D adalah suatu Port I/O 8-bit dua arah dengan resistor *internal pull-up* (yang dipilih untuk beberapa bit). Port D *output buffer* mempunyai karakteristik

gerakan simetris dengan keduanya *sink* tinggi dan kemampuan sumber. Pin Port D juga merupakan *tri-stated* jika kondisi *reset* menjadi aktif, sekalipun waktu habis.

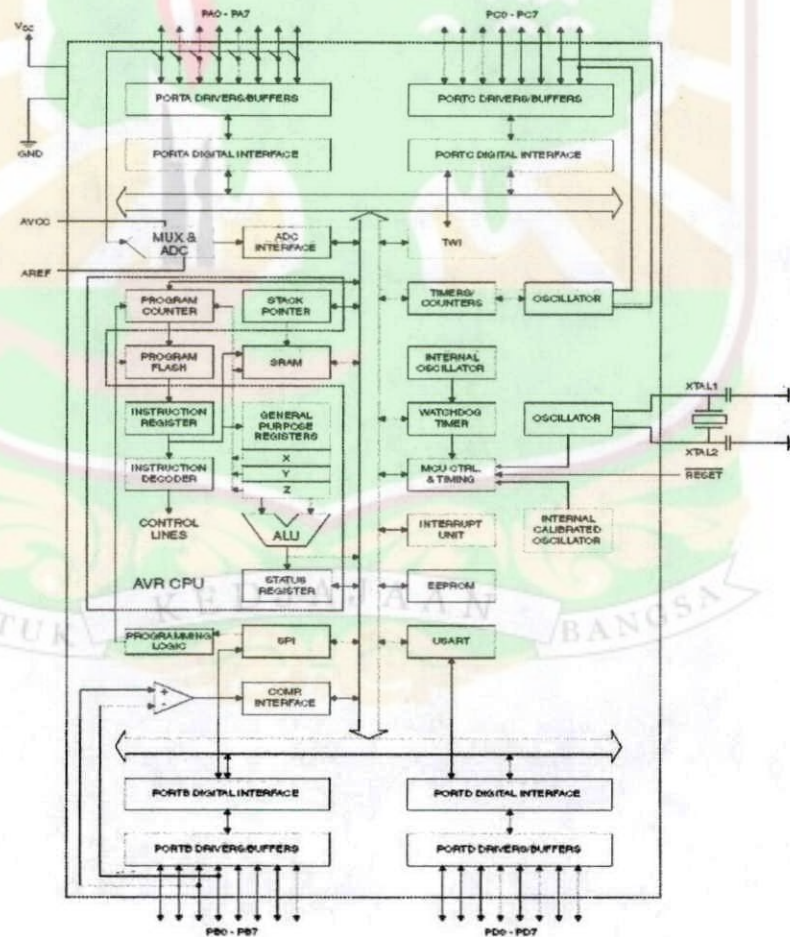
- RESET (*Reset input*)
- XTAL1 (*Input Oscillator*)
- XTAL2□ (*Output Oscillator*) AVCC adalah pin penyedia tegangan untuk port

A dan A/D konverter

AREF adalah pin referensi analog untuk A/D konverter.

2.4.1.2 Sistem Minimum Mikrokontroler ATMEGA 8535

Gambar 2.13 menunjukkan arsitektur mikrokontroler ATMEGA 8535.



Gambar 2.13 Arsitektur Mikrokontroler ATMEGA 8535

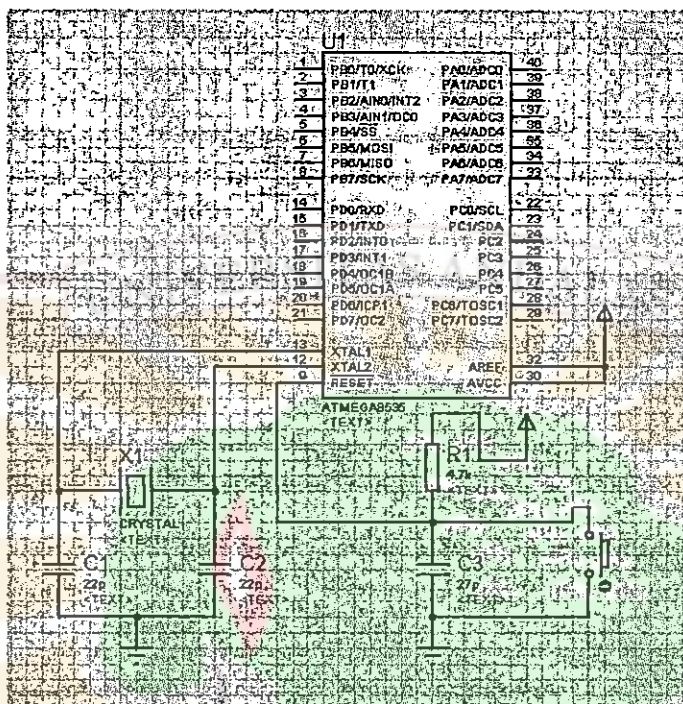
Secara garis besar, arsitektur mikrokontroler ATMEGA8535 terdiri dari :

- * 32 saluran I/O (Port A, Port B, Port C, dan Port D)
- * 10 bit 8 Channel ADC (Analog to Digital Converter)
- * 4 channel PWM
- * 6 Sleep Modes : Idle, ADC Noise Reduction, Power-save, Power-

down, Standby and Extended Standby

- * 3 buah timer/counter
- * Analog comparator
- * Watchdog timer dengan osilator internal
- * 512 byte SRAM
- * 512 byte EEPROM
- * 8 kb Flash memory dengan kemampuan Read While Write
- * Unit interupsi (internal & eksternal)
- * Port antarmuka SPI8535 “memory map”
- * Port USART untuk komunikasi serial dengan kecepatan maksimal 2,5Mbps
- * 4.5 sampai 5.5V operation, 0 sampai 16MHz

Gambar 2.14 menunjukkan Sistem minimum mikrokontroler AT Mega 8535.



Gambar 2.14 Sistem Minimum mikrokontroler AT Mega 8535.

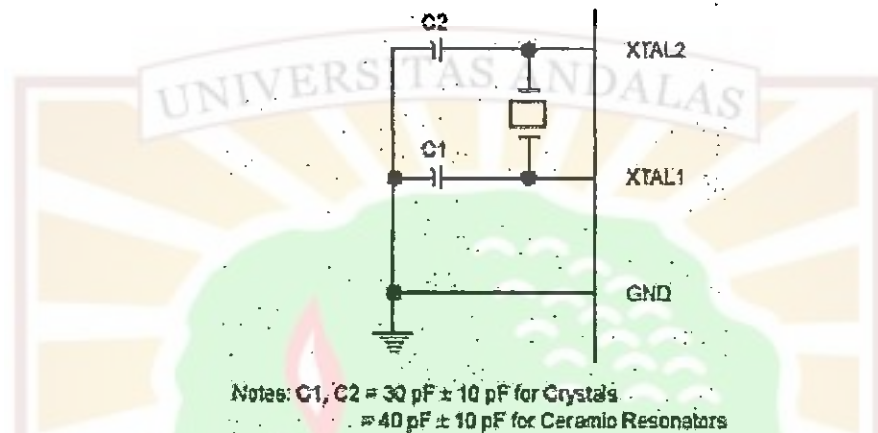
Sistem minimum mikrokontroler ATMEGA 8535 bekerja sebagai berikut : kapasitor C3 dan resistor R1 digunakan untuk sistem *reset*, saat pertama suplay diberikan ke mikrokontroler maka kaki 9 akan berlogika 1, selama 2 siklus mesin. Setelah itu pin 9 akan berlogika 0 kembali. Proses seperti ini bisa terjadi berdasarkan proses pengisian dan pengosongan kapasitor.

Kapasitor C1 dan C2, dipasang bersamaan dengan keramik resonator (x-tal) untuk menghasilkan *clock* internal. Nilai dari *clock* ini tergantung dari keramik resonator (X-TAL) yang diberikan.

2.4:1.3 Sistem Clock

Mikrokontroler mempunyai sistem pewaktuan CPU, 12 siklus *clock*. Artinya setiap 12 siklus yang dihasilkan oleh *ceramic resonator* maka akan menghasilkan

satu siklus mesin. Nilai ini yang akan menjadi acuan waktu operasi CPU. Untuk mendesain sistem mikrokontroler kita memerlukan sistem *clock*, sistem ini bisa dibangun dari *clock* eksternal maupun *clock* internal. Untuk *clock* internal, pemasangan komponen seperti pada gambar 2.15.



Gambar 2.15 Sistem Clock

2.4.1.4 Organisasi Memori AVR ATmega8535

AVR ATmega8535 memiliki ruang pengalamatan memori data dan memori program yang terpisah. Sebagai tambahan, ATmega8535 memiliki fitur suatu EEPROM memori untuk penyimpanan data.

Memori Data

Memori data terbagi menjadi 3 bagian, yaitu 32 register umum, 64 register I/O, dan 512 byte SRAM internal. Register umum menempati *space* data pada alamat terbawah, yaitu \$00 sampai \$1F. Sementara itu, register khusus untuk menangani I/O dan kontrol terhadap mikrokontroler menempati 64 alamat berikutnya, yaitu mulai dari \$20 hingga \$5F. Register tersebut merupakan register yang khusus digunakan untuk mengatur fungsi terhadap berbagai *peripheral* mikrokontroler, seperti kontrol register, *timer/counter*, fungsi – fungsi I/O, dan

sebagainya. Alamat memori berikutnya digunakan untuk SRAM 512 *byte*, yaitu pada lokasi \$60 sampai dengan \$25F. Konfigurasi memori data ditunjukkan pada gambar 2.16.

Register Umum		Alamat
R0		\$0000
R1		\$0001
...		
R30		\$001E
R31		\$001F

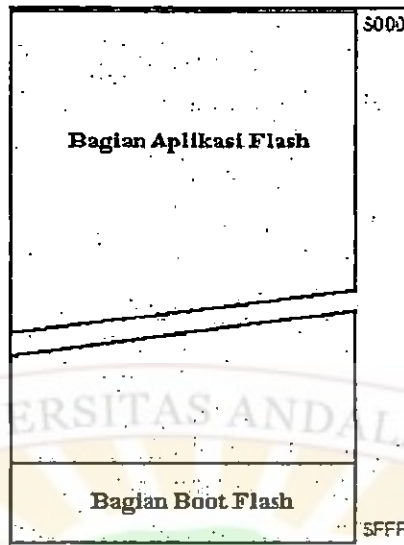
Register I/O		SRAM Internal
\$00		\$0020
\$01		\$0021
...		
\$3E		\$005E
\$3F		\$005F

\$0060
\$0061
.....
\$025E
\$025F (RAMEND)

Gambar 2.16 Konfigurasi Data AVR AT Mega 8535

Memori Program

ATmega8535 berisi 8Kbytes *on-chip* di dalam sistem *Memori Flash Reprogrammable* untuk penyimpanan program. Karena semua AVR instruksi adalah 16 atau 32 bits lebar. *Flash* adalah berbentuk 4K x16. Untuk keamanan perangkat lunak, *flash* ruang program memori adalah dibagi menjadi dua bagian, bagian *boot* program dan bagian aplikasi program dengan alamat mulai dari \$000 sampai \$FFF. *Flash* memori mempunyai suatu daya tahan sedikitnya 10,000 *write/erase cycles*. Program Counter (PC) adalah 12 bit lebar, alamat ini 4K lokasi program memori. Peta memori AT Mega 8535 dapat dilihat pada gambar 2.17.



Gambar 2.17 Peta Memori Program AT Mega 8535

Port Sebagai *Input / Output* Digital

ATmega8535 mempunyai empat *port* yaitu PortA, PortB, PortC, dan PortD. Keempat *port* tersebut merupakan jalur *bi-directional* dengan pilihan *internal pull-up*. Tiap *port* mempunyai tiga buah register bit, yaitu DDxn, PORTxn, dan PINxn. Huruf 'x' mewakili nama huruf dari *port* sedangkan huruf 'n' mewakili nomor bit. Bit DDxn terdapat pada I/O address DDRx, bit PORTxn terdapat pada I/O address PORTx, dan bit PINxn terdapat pada I/O address PINx. Bit DDxn dalam register DDRx (*Data Direction Register*) menentukan arah pin. Bila DDxn diset 1 maka Px berfungsi sebagai pin *output*. Bila DDxn diset 0 maka Px berfungsi sebagai pin *input*. Bila PORTxn diset 1 pada saat pin terkonfigurasi sebagai pin *input*, maka resistor *pull-up* akan diaktifkan. Untuk mematikan resistor *pull-up*, PORTxn harus diset 0 atau pin dikonfigurasi sebagai pin *output*. Pin *port* adalah *tri-state* setelah kondisi *reset*. Bila PORTxn diset 1 pada saat pin terkonfigurasi sebagai pin *output* maka pin *port* akan berlogika 1. Dan bila PORTxn diset 0 pada saat pin terkonfigurasi sebagai pin *output* maka pin *port*

akan berlogika 0. Saat mengubah kondisi *port* dari kondisi *tri-state* (DDxn=0, PORTxn=0) ke kondisi *output high* (DDxn=1, PORTxn=1) maka harus ada kondisi peralihan apakah itu kondisi *pull-up enabled* (DDxn=0, PORTxn=1) atau kondisi *output low* (DDxn=1, PORTxn=0). Biasanya, kondisi *pull-up enabled* dapat diterima sepenuhnya, selama lingkungan impedansi tinggi tidak memperhatikan perbedaan antara sebuah *strong high driver* dengan sebuah *pull-up*. Jika ini bukan suatu masalah, maka bit PUD pada register *SFIOR* dapat diset 1 untuk mematikan semua *pull-up* dalam semua port. Peralihan dari kondisi *input dengan pull-up* ke kondisi *output low* juga menimbulkan masalah yang sama. Maka harus menggunakan kondisi *tri-state* (DDxn=0, PORTxn=0) atau kondisi *output high* (DDxn=1, PORTxn=0) sebagai kondisi transisi. Konfigurasi pin-pin pada port ATMEGA 8535 dapat dilihat pada table 2.3.

Tabel 2.3 Konfigurasi Pin Port pada mikrokontroler ATMEGA 8535

DDxn	PORTxn	PUD (in SFIOR)	I/O	Pull-up	Comment
0	0	X	Input	No	Tri-state (Hi-Z)
0	1	0	Input	Yes	Pxn will source current if ext. pulled low.
0	1	1	Input	No	Tri-state (Hi-Z)
1	0	X	Output	No	Output Low (Sink)
1	1	X	Output	No	Output High (Source)

Port-port mikrokontroler dilengkapi *pull up internal*, sehingga kondisi *default*-nya adalah *high*. Untuk menjadikan port sebagai *input*, maka kondisinya adalah logika *high* atau membiarkannya dalam kondisi *default*. Pada prinsip kerja *hardware*, saat menekan tombol SW maka *signal* rendah akan dikirim ke *port* pada mikrokontroler. Dengan demikian, port akan mempunyai logika sesuai dengan penekanan tombol.

2.4.2 Motor DC

Motor DC adalah alat yang dapat mengubah daya listrik DC menjadi daya mekanik. Apabila pada penghantar yang dialiri listrik dan terletak diantara dua buah kutub magnet (kutub utara dan kutub selatan) maka pada penghantar tersebut akan terjadi gaya yang menggerakkan penghantar tersebut. Suatu kumparan yang terletak dalam medan magnet yang arah arus dari kedua sisinya berlawanan sehingga arah gerak terhadap putaran berbeda selanjutnya akan menghasilkan gaya gerak putar atau kopel. Semakin besar arusnya maka akan semakin besar kopelnya, juga jika gaya magnetnya makin kuat kopelnya makin berat. Jika kumparan terletak diantara kutub magnet yang sedang berputar maka pada kumparan tersebut akan timbul suatu tegangan dari luar yang disebut gaya gerak listrik (ggl) lawan. Besar kecilnya ggl lawan tergantung dari tahanan jangkarnya.

Konstruksi motor DC terdiri atas beberapa bagian yang meliputi badan motor, inti kutub magnet, sikat-sikat, komutator, dan jangkar. Gambar motor DC seperti ditunjukkan pada Gambar 2.18.



Gambar 2.18 Bagian Motor Arus Searah

a. Badan Motor

Fungsi utama badan motor adalah sebagai bagian dari tempat mengalirnya fluks magnet yang dihasilkan kutub-kutub magnet, oleh karena itu badan motor terbuat dari bahan *ferromagnetic*.

b. Inti kutub magnet

Inti kutub magnet berfungsi untuk menghasilkan fluksi magnet. Magnet yang dipakai adalah magnet permanen jadi di sini tidak menggunakan prinsip elektromagnetik.

c. Sikat-sikat

Fungsi dari sikat-sikat adalah sebagai jembatan bagi aliran arus listrik dari sumber tegangan ke lilitan jangkar. Di samping itu sikat-sikat memegang peranan penting dalam terjadinya komutasi. Agar gesekan-gesekan antara komutator dan sikat tidak mengakibatkan ausnya komutator maka sikat harus lebih lunak dari komutator yang biasanya dibuat dari bahan arang (*coal*).

d. Komutator

Komutator berfungsi sebagai penyearah mekanik, yang bersama-sama dengan sikat-sikat membuat suatu kerjasama yang disebut komutasi. Agar menghasilkan penyearahan yang lebih baik atau lebih rata maka komutator yang digunakan berjumlah cukup besar.

e. Jangkar

Jangkar yang umumnya digunakan dalam motor arus searah adalah jangkar yang berbentuk silinder yang diberi alur-alur pada permukaannya untuk tempat melilitkan kumparan-kumparan.

2.4.3 Motor Stepper

Motor stepper adalah perangkat elektromekanis yang bekerja dengan mengubah pulsa elektronis menjadi gerakan mekanis diskrit. Motor stepper bergerak berdasarkan urutan pulsa yang diberikan kepada motor. Karena itu, untuk menggerakkan motor stepper diperlukan pengendali motor stepper yang membangkitkan pulsa-pulsa periodik. Penggunaan motor stepper memiliki

beberapa keunggulan dibandingkan dengan penggunaan motor DC biasa.

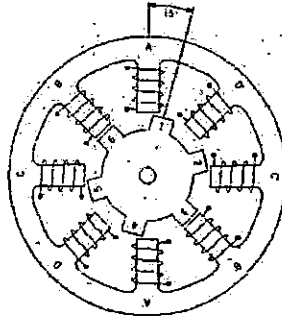
Keunggulannya antara lain adalah :

- Sudut rotasi motor proporsional dengan pulsa masukan sehingga lebih mudah diatur
- Motor dapat langsung memberikan torsi penuh pada saat mulai bergerak
- Posisi dan pergerakan repetisinya dapat ditentukan secara presisi
- Memiliki respon yang sangat baik terhadap mulai, stop dan berbalik (perputaran)
- Sangat realibel karena tidak adanya sikat yang bersentuhan dengan rotor seperti pada motor DC
- Dapat menghasilkan perputaran yang lambat sehingga beban dapat dikopel langsung ke porosnya
- Frekuensi perputaran dapat ditentukan secara bebas dan mudah pada range yang luas.

Pada dasarnya terdapat 3 tipe motor stepper yaitu:

1. Motor stepper tipe Variable reluctance (VR).

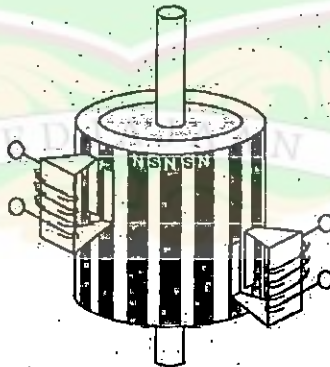
Motor stepper jenis ini telah lama ada dan merupakan jenis motor yang secara struktural paling mudah untuk dipahami. Motor ini terdiri atas sebuah rotor besi lunak dengan beberapa gerigi dan sebuah lilitan stator. Ketika lilitan stator diberi energi dengan arus DC, kutub-kutubnya menjadi termagnetasi. Perputaran terjadi ketika gigi-gigi rotor tertarik oleh kutub-kutub stator. Gambar 2.19 adalah penampang melintang dari motor stepper tipe variable reluctance (VR):



Gambar 2.19 Penampang Melintang Dari Motor Stepper Tipe Variable Reluctance (VR).

2. Motor stepper tipe Permanent Magnet (PM).

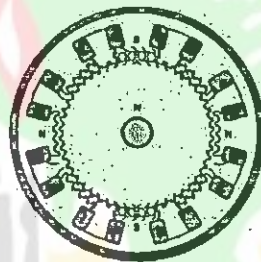
Motor stepper jenis ini memiliki rotor yang berbentuk seperti kaleng bundar (*tin can*) yang terdiri atas lapisan magnet permanen yang diselang-seling dengan kutub yang berlawanan. Dengan adanya magnet permanen, maka intensitas fluks magnet dalam motor ini akan meningkat sehingga dapat menghasilkan torsi yang lebih besar. Motor jenis ini biasanya memiliki resolusi langkah (step) yang rendah yaitu antara 7,50 hingga 150 per langkah atau 48 hingga 24 langkah setiap putarannya. Gambar 2.20 adalah ilustrasi sederhana dari motor stepper tipe permanent magnet.



Gambar 2.20 Motor Stepper Tipe Permanent Magnet.

3. Motor stepper tipe *Hybrid* (HB).

Motor stepper tipe *hybrid* memiliki struktur yang merupakan kombinasi dari kedua tipe motor stepper sebelumnya. Motor stepper tipe *hybrid* memiliki gigi-gigi seperti pada motor tipe VR dan juga memiliki magnet permanen yang tersusun secara aksial pada batang porosnya seperti motor tipe PM. Motor tipe ini paling banyak digunakan dalam berbagai aplikasi karena kinerja lebih baik. Motor tipe hibrid dapat menghasilkan resolusi langkah yang tinggi yaitu antara 3,60 hingga 0,90 per langkah atau 100-400 langkah setiap putarannya. Gambar 2.21 adalah penampang melintang dari motor stepper tipe *hybrid*.



Gambar 2.21 Penampang Melintang Dari Motor Stepper Tipe *Hybrid*.

BAB III

METODELOGI PENELITIAN

3.1 Jenis Penelitian

Berdasarkan tujuan dan manfaat tugas akhir ini maka penelitian ini termasuk ke dalam jenis penelitian terapan, maksudnya bahwa penelitian ini merupakan pemecahan terhadap suatu masalah untuk tujuan tertentu dan merupakan aplikasi baru dari penelitian yang telah ada dengan berpedoman pada data sekunder (data dari hasil penelitian) yang relevan. Sedangkan bila ditinjau dari sifat-sifat masalahnya maka penelitian ini bersifat eksperimen.

3.2 Sampel Penelitian

Sampel penelitian yang digunakan adalah pergerakan 1 buah *prototipe* mesin pengaduk kopi otomatis. Penelitian ini menggunakan air kopi yang dimasukkan ke dalam gelas bening. *Webcam* yang ditempatkan tepat di depan gelas akan menangkap video secara *real time* sehingga saat pengambilan sampel warna kopi, tutup saluran krimer dan pengaduk kopi bergerak sesuai dengan pilihan jenis kopi. Ketika saluran krimer terbuka, krimer akan jatuh ke dalam cairan kopi, menyebabkan perubahan warna pada kopi yang akan ditangkap oleh kamera.

Sampel penelitian yang diambil terdiri atas tiga kondisi, yaitu

1. Pada saat warna kopi yang ditangkap telah sesuai dengan jenis kopi yang dipilih pada *input*, maka pengaduk kopi akan berhenti dan penutup saluran krimer akan tertutup.
2. Pada saat warna kopi yang ditangkap belum sesuai dengan jenis kopi yang dipilih pada *input*, maka pengaduk kopi akan tetap bergerak dan penutup saluran krimer akan terbuka.
3. Pada saat jenis kopi yang dipilih pada *input* adalah “Sangat Pekat/*Very Dark*”, maka hanya pengaduk kopi yang bergerak. Penutup saluran krimer akan tetap diam.

Penelitian ini menggunakan objek secara langsung, yaitu ketika *webcam* dalam keadaan *capture*, *webcam* akan mendeteksi warna kopi, dimana saat terjadi perubahan warna pada citra kopi, dalam hal ini disebabkan oleh krimer, maka pengaduk kopi dan penutup saluran krimer akan bergerak atau berhenti tergantung kecocokan antara jenis kopi yang dipilih pada *input* dengan warna kopi yang ditangkap oleh kamera. Pergerakan pengaduk kopi dan penutup saluran krimer dapat diamati secara langsung.

3.3 Teknik analisa sistem

Kinerja sistem yang akan dieksperimenkan dianalisa dengan menggunakan teknik analisa objektif. Analisa ini diperoleh dari hasil pengamatan langsung terhadap visualisasi *output* sistem berupa pergerakan pengaduk kopi dan penutup saluran krimer sesuai dengan warna citra kopi yang di *capture* oleh *webcam*. Visualisasi juga ditampilkan dalam bentuk GUI (*Graphical User Interface*) MATLAB, kemudian hasil-hasil yang

didapatkan melalui eksperimen (*output* sistem) akan dibandingkan dengan *input* sistem.

3.4 Prosedur Penelitian

Dalam penelitian ini digunakan langkah-langkah seperti berikut :

1. Studi Literatur

Studi literatur dilakukan untuk mendapatkan pemahaman tentang konsep-konsep teoritis yang berhubungan dengan tema penelitian. Bahan-bahan yang menjadi sumber referensi untuk studi literatur didapatkan melalui internet dan buku-buku panduan yang terkait.

2. Diskusi dengan pembimbing

3. Perancangan algoritma dan pembuatan program

Program yang dirancang dalam tugas akhir ini terbagi atas 7 (tujuh) algoritma pokok, yaitu:

1. Algoritma ekstraksi nilai RGB citra. Algoritma ini bertujuan untuk mendapatkan nilai RGB citra.
2. Algoritma konversi ruang warna RGB ke HIS. Algoritma ini bertujuan untuk mendapatkan nilai *Hue*, *Saturation* dan *Intensity* dari citra.
3. Algoritma pengambilan keputusan. Algoritma ini merupakan *Fuzzy Inference System* (FIS) .Pada penelitian ini, FIS yang digunakan adalah model Mamdani dan Sugeno.
4. Algoritma pengiriman data ke mikrokontroler menggunakan *parallel port*.

Berdasarkan algoritma yang telah ditentukan di atas, dibuat program pengolahan *frame-frame* video dengan sebuah *webcam* menggunakan bahasa pemrograman MATLAB dan bahasa C pada mikrokontroler.

5. Perancangan antarmuka sistem

Pembuatan sistem antarmuka dilakukan untuk dapat memposisikan motor DC sebagai penggerak pengaduk kopi dan motor stepper sebagai penggerak penutup katup krimer dengan komputer sebagai pengolah data, dimana kesamaan pergerakan motor DC dan motor stepper terhadap warna citra kopi pada *frame* video dapat dijadikan parameter dari keberhasilan program yang telah dibuat.

6. Pengujian terhadap hasil

Pengujian dilakukan dengan cara mengamati pergerakan pengaduk kopi dan penutup saluran krimer yang telah diposisikan dengan motor DC dan motor stepper, apakah sudah sama dengan pergerakan pengaduk kopi dan penutup saluran krimer yang diinginkan.

7. Analisa keluaran sistem

Analisa keluaran sistem didasarkan pada teknik analisa sistem yang telah dijabarkan pada subbab 3.3. Keluaran dari sistem ini adalah berupa pergerakan pengaduk kopi dan penutup saluran krimer. Sedangkan visualisasinya pada program aplikasi GUI MATLAB adalah berupa visualisasi dan keterangan hasil penarikan kesimpulan menggunakan *Fuzzy Inference System* yang menyatakan jenis kopi, serta aksi yang akan

dilakukan pengaduk kopi dan penutup saluran krimer sesuai jenis kopi tersebut.

8. Penyusunan laporan penelitian

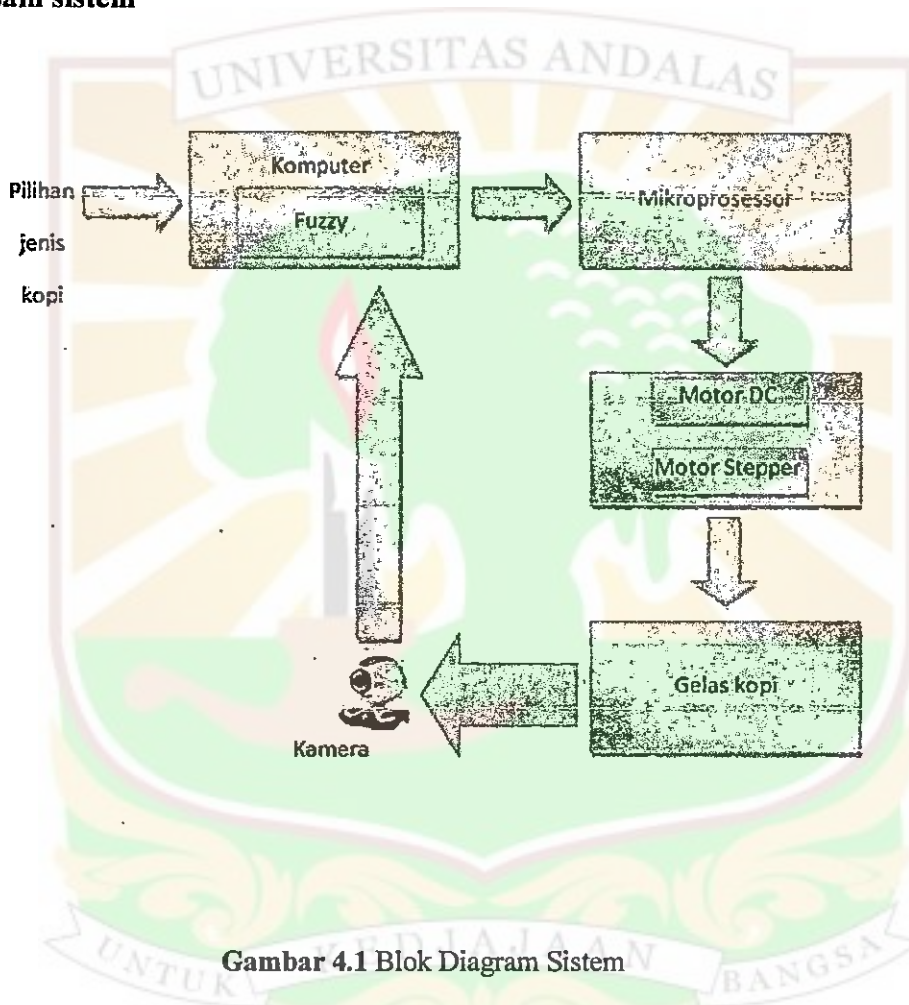
Langkah terakhir dari penelitian ini adalah penyusunan laporan hasil penelitian. Hal ini dilakukan setelah semua tahapan dilalui dan diperoleh hasil akhir.



BAB IV

PERANCANGAN SISTEM IMPLEMENTASI FUZZY LOGIC DAN SENSOR WARNA PADA MESIN PENGADUK KOPI OTOMATIS

4.1 Desain sistem



Gambar 4.1 Blok Diagram Sistem

Perancangan sistem secara umum ditunjukkan pada gambar 4.1. Perancangan sistem menggambarkan cara kerja sistem secara umum.

Sistem dimulai dengan input dari *user*. *User* memilih jenis kopi yang diinginkan dengan mengklik salah satu tombol jenis kopi yang tersedia pada GUI (*Graphical User Interface*). Setelah tombol diklik, komputer akan mengirim data ke mikrokontroler

melalui port paralel. Berdasarkan data yang diterima, mikrokontroler akan menggerakkan motor dc dan motor stepper untuk mengaduk kopi dan membuka katup corong krimer. Sementara kopi di dalam gelas diaduk oleh pengaduk kopi, kamera yang diletakkan di depan gelas kopi akan meng-*capture* citra kopi di dalam gelas dan mengirimkannya ke komputer untuk diolah. Di komputer, citra kopi tersebut akan diolah untuk mendapatkan nilai RGB dari citra. Nilai RGB tersebut akan dikonversi menjadi nilai HIS. Nilai HIS tersebut merupakan input bagi sistem *Fuzzy Inference System* (FIS) untuk menentukan jenis kopi yang tertangkap oleh kamera. Keluaran dari FIS adalah jenis kopi yang tertangkap oleh kamera. Apabila telah terjadi kecocokan antara jenis kopi yang diminta oleh *user* dengan keluaran FIS, maka komputer akan mengirim data ke mikrokontroler dan mikrokontroler akan menghentikan gerakan pengaduk kopi dan menutup katup corong krimer. Apabila belum terjadi kecocokan antara jenis kopi yang diminta oleh *user* dengan keluaran FIS, maka komputer tidak akan mengirimkan data apapun ke mikrokontroler dan akan memerintahkan kamera untuk meng-*capture* kembali gambar kopi dalam gelas.

4.2 Perancangan perangkat lunak (*software*).

4.2.1 Membuat video objek (akuisisi data).

Penelitian ini dimulai dengan proses penangkapan (*capture*) video digital dengan membuat video objek menggunakan *web camera* berupa *frame-frame image*. Proses yang terjadi adalah ketika kamera ditujukan pada objek yang akan diambil, maka benda tersebut akan memantulkan cahaya ke kamera sebesar intensitas cahaya yang diterima benda dikalikan dengan cahaya yang dipantulkan benda terhadap kamera. Objek yang ditangkap kamera selanjutnya

diolah melalui program. Pada MATALAB 7.0.1 sudah disediakan *image aquisition toolbox* dengan fungsi *videoinput* untuk mengakuisisi citra atau video. Video objek ini hanya disimpan dalam *memory* (RAM) selama pengolahan oleh program.

Baik atau tidaknya hasil tangkapan *image* oleh kamera tergantung dari spesifikasi internal dari kamera itu sendiri. Pada penelitian ini digunakan kamera *webcam* dengan spesifikasi kamera sebagai berikut:

Merek	: Mediatech USB Digital PC Camera-003
Video Format	: 24-bit RGB
Interface	: USB 1.1
Frame Rate	: 320x240 up to 30 frame/sec (CIF) 640x480 up to 15 frame/sec (VGA)
Sensor Size	: 4.86x3.64 mm ²
S/N Ratio	: <48dB
Dynamic range	: <72dB
Focus range	: 5cm – infinity

Gambar 4.2 menunjukkan *webcam* yang digunakan dalam penelitian.



Gambar 4.2 Webcam

4.2.2 Konversi Ruang Warna Citra

Pada sistem ini, program akan mengekstrak nilai RGB dari citra. Nilai RGB ini akan dikonversikan ke ruang warna HIS (*Hue, Intensity, Saturation*). Nilai *hue* citra akan digunakan sebagai input sistem Fuzzy, untuk menentukan jenis kopi yang ditangkap oleh kamera.

4.2.3 Penggunaan *Fuzzy Logic Toolbox* dalam Pembentukan *Database* Jenis Kopi pada Sistem Penarik Kesimpulan

Proses pembentukan *database* jenis kopi pada sistem *Fuzzy* dalam penelitian ini menggunakan bantuan *Fuzzy Logic Toolbox* yang terdapat di dalam *software* MATLAB. *Fuzzy Logic Toolbox* ini merupakan sekumpulan *tool* yang berbentuk *Graphical User Interface* (GUI) yang dapat digunakan dalam merancang sistem *Fuzzy*. Dengan menggunakan *toolbox* ini, dapat dibangun sebuah *Fuzzy Inference System* (FIS) dalam lingkungan kerja MATLAB. Dalam *toolbox* ini, terdapat dua model FIS yang digunakan, yaitu FIS model Mamdani dan FIS model Sugeno. Kedua model FIS tersebut digunakan dalam penelitian ini.

Proses pembentukan sistem *Fuzzy* dalam *toolbox* ini dimulai dengan penentuan model FIS dan penentuan operator *Fuzzy* yang digunakan melalui FIS *Editor*, kemudian fuzzifikasi yaitu pembentukan *Membership Function Input* dan *Membership Function Output* melalui *Membership Function Editor* lalu dilanjutkan dengan pembentukan *rule* melalui *Rule Editor* dan *Rule Viewer* untuk menampilkan proses keseluruhan yang terjadi dalam FIS yang telah dibangun. Pada penelitian ini, rancangan sistem FIS menggunakan dua variabel

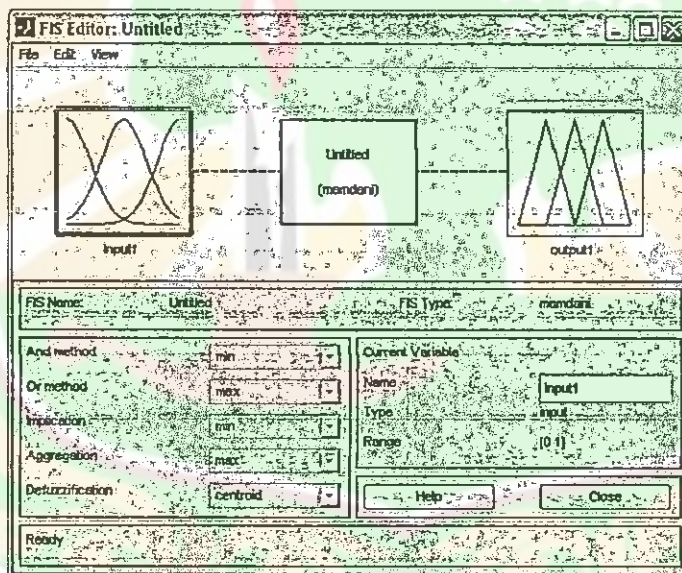
input dan satu variabel *output*. Variabel *input* yang digunakan adalah nilai *hue* dan *intensity* dari citra kopi. Untuk variable *output* yang digunakan jenis kopi yang dikenali, yaitu “Sangat Pekat/*Very Dark*”, “Pekat/*Dark*”, “Biasa/*Middle*”, “Ringan/*Light*” dan “Sangat Ringan/*Very Light*”.

4.2.3.1 FIS Editor

Untuk mulai menggunakan FIS Editor pada MATLAB Prompt, masukkan perintah :

```
>> fuzzy
```

Maka akan muncul FIS Editor seperti tampak pada Gambar 4.3.

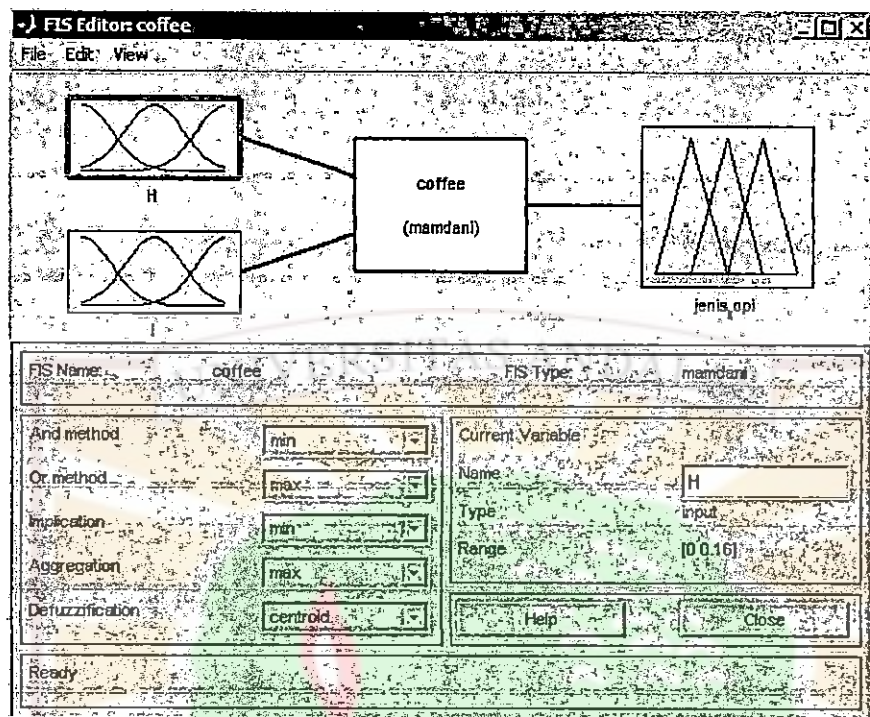


Gambar 4.3 FIS Editor

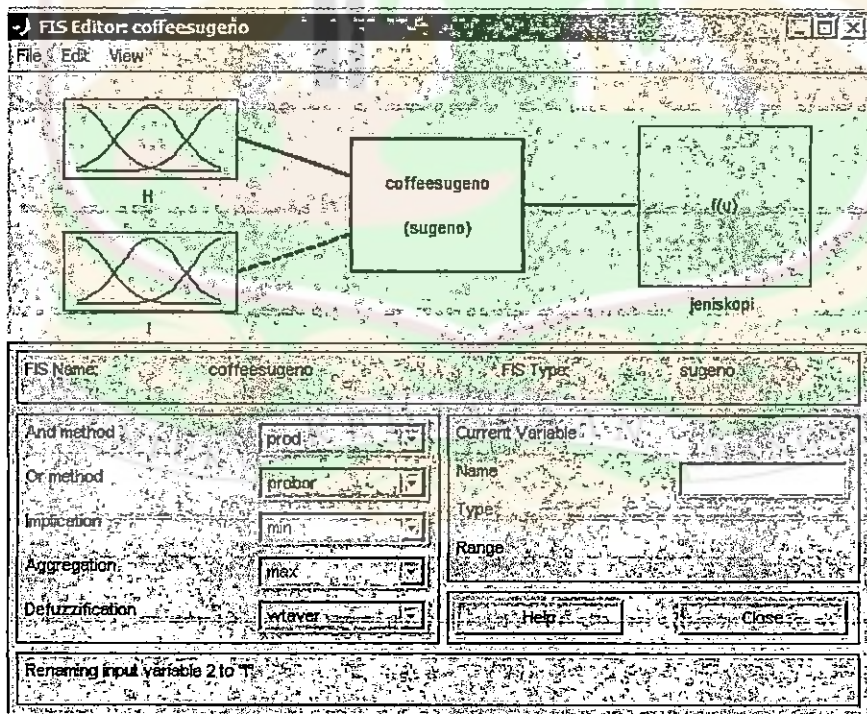
Pada penelitian ini, tipe FIS yang digunakan adalah FIS Mamdani dan FIS Sugeno dengan satu variabel masukan, yaitu *H*, dan satu variabel keluaran yaitu *Jenis_Kopi*. Langkah-langkah yang dilakukan untuk membangun FIS tipe Mamdani dan Sugeno pada umumnya sama, yaitu :

1. Pilih menu *file*
2. Ubah label *input1* menjadi *H* dengan mengklik kotak kuning berlabel *input1* (saat diklik batas kotak akan berubah warna menjadi merah), lalu pada bagian *Current variable* disebelah kanan bawah, pada kolom Nama, variabel *input1* diganti menjadi *H* dan tekan *Enter*.
3. Ubah label *output1* menjadi *JenisKopi* dengan mengklik kotak biru di kanan atas (saat di klik, batas kotak akan berubah warna menjadi merah). Lalu pada bagian *Current Variable* disebelah kanan bawah, pada kolom Nama, ganti variabel *output1* menjadi *JenisKopi* dan tekan *Enter*.
4. Pilih menu *File > Export > To Disk* lalu ubah nama *file* menjadi *coffee*. Jadi, ketika ingin membuka kembali FIS yang telah disimpan cukup mengetikkan *fuzzy coffee* pada *MATLAB Prompt*.
5. Pada pembuatan FIS Sugeno, langkah-langkah yang dilakukan sama seperti dijelaskan pada langkah satu hingga lima. Namun nama *file* diubah menjadi *coffeesugeno*.

Setelah semua langkah di atas selesai dilakukan maka dapat dilihat sebuah diagram FIS dengan label masukan dan keluaran yang telah dibuat seperti ditunjukkan pada gambar 4.4 dan gambar 4.5.



Gambar 4.4 Diagram FIS *coffee*



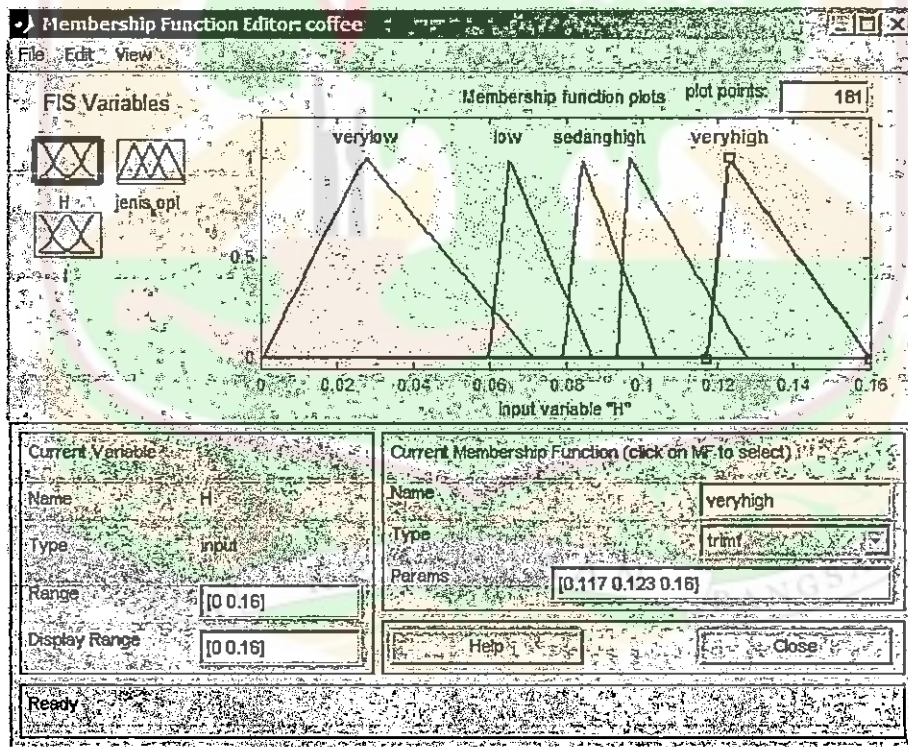
Gambar 4.5 Diagram FIS *coffeesugeno*

4.2.3.2 Membership Function Editor

Fungsi-fungsi keanggotaan variabel masukan dan keluaran didefinisikan melalui *Membership Function Editor*. Untuk membuka *Membership Function Editor* ini dapat dilakukan dengan cara sebagai berikut :

1. Dari *FIS Editor*, pilih *Edit > Membership Functions*.
2. Dari *FIS Editor*, klik ganda ikon variabel masukan / keluaran.

Dari *MATLAB Prompt*, ketikkan *mfedit coffee* jika diasumsikan *FIS Editor* sebelumnya telah ditutup. Maka dapat dilihat *Window Membership Function Editor* seperti pada Gambar 4.6 berikut:



Gambar 4.6 *Membership Function Editor*

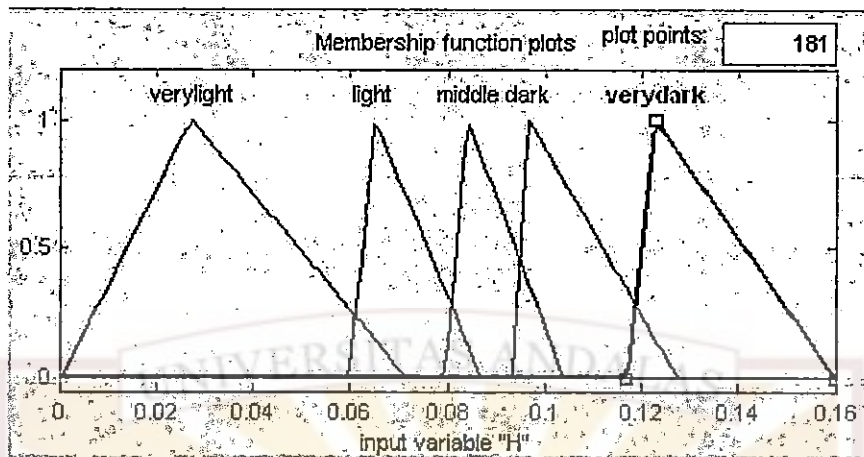
Masukan yang digunakan pada *Membership Function Editor* ini merupakan nilai *hue* citra kopi, pada *H* dalam variabel *input*. Dalam satu

variabel *input* terdapat lima *Membership Function* yang mewakili besar rata-rata nilai dalam karakteristik variabel *input*. Kelima *Membership Function* tersebut yaitu *verylight*, *light*, *middle*, *dark* dan *verydark*. Sedangkan pada variabel keluaran terdapat lima *Membership Function* yang mewakili jenis kopi Sangat Pekat, Pekat, Biasa, Ringan, dan Sangat Ringan.

Berikut ini langkah-langkah dalam pembentukan *Membership Function* variabel masukan *H*:

1. Klik ikon variabel masukan *H*.
2. Pada bagian kiri bawah, ubah *field Range* menjadi [0 0.16], lalu tekan *Enter*.
Nilai *range* ini merupakan rentang dari nilai hue dalam bilangan *double*.
3. Pada bagian *field Display Range*, nilainya akan mengikuti nilai pada *Field Range*.
4. Klik kurva yang berlabel *mf1*. Pada bagian *field Name*, ubah *mf1* menjadi *verydark*, lalu *mf2* menjadi *dark*, *mf3* menjadi *middle*, *mf4* menjadi *light*, dan *mf5* menjadi *verylight* lalu tekan *Enter*.
5. Pada bagian *field Type*, klik panah bawah lalu pilih opsi tipe fungsi keanggotaan *trimmf* untuk label *verydark*, *dark*, *middle*, *light* dan *verylight*. Pada bagian *field Params*, nilainya akan berubah-ubah secara otomatis ketika kurva pada setiap *Membership Function* digeser.
6. Pada pembuatan *membership function* FIS Sugeno, langkah-langkah yang dilakukan pada pembuatan *membership function* Mamdani.

Contoh hasil dari penggunaan *Membership Function* di atas ditunjukkan dalam Gambar 4.7 berikut:



Gambar 4.7 Hasil Pembentukan Fungsi Keanggotaan Melalui *Membership Function editor*

Untuk *Membership Function* keluaran, langkah yang dilakukan sama dengan langkah yang dilakukan dalam pembentukan *Membership Function* variabel masukan, hanya saja **mf1** sampai **mf3** dirubah menjadi *verydark*, *dark*, *middle*, *light*, dan *verylight*.

4.2.3.3 Rule Editor.

Proses pembentukan *rule* dalam *Fuzzy Logic toolbox* ini dapat dilakukan pada bagian *Rule Editor*. Untuk membuka *Rule Editor* ini, pada jendela *FIS Editor* klik menu **Edit > Rules**. Pada proses evaluasi *rule* ini ditetapkan aturan yang menetapkan bagaimana warna kopi dapat dikenali. Berdasarkan data yang diperoleh dari hasil ekstraksi nilai *hue* pada jenis kopi yang diujikan dan *Membership Function* untuk variabel *input* telah terbentuk, maka dapat dibuat *rule* atau aturan bagaimana jenis kopi tersebut dikenali.

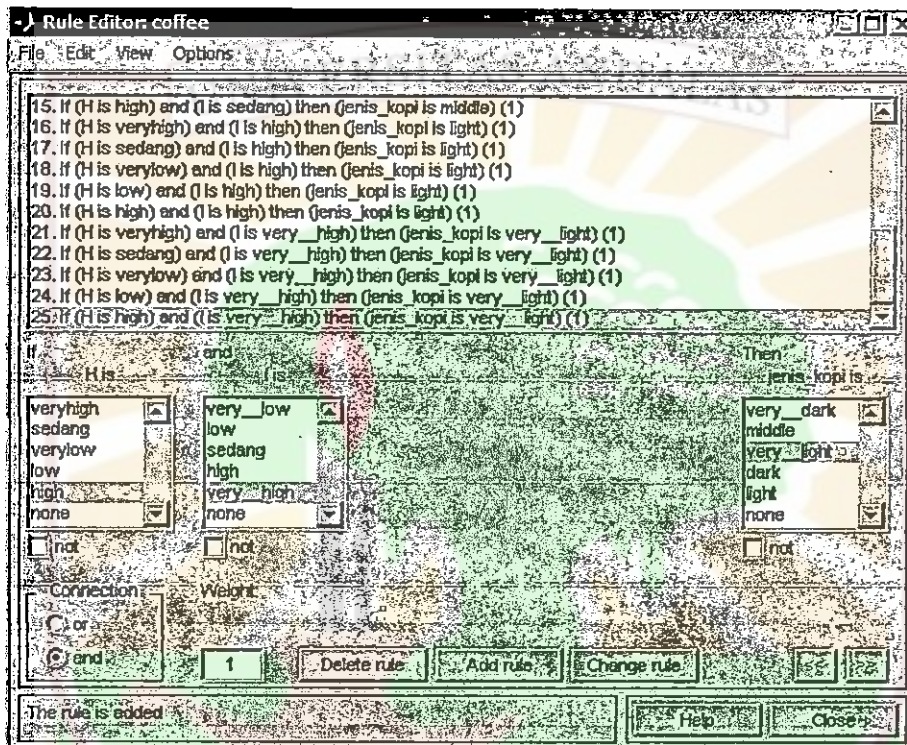
Untuk membuat sebuah *rule*, langkah-langkahnya sebagai berikut :

1. Di bawah variabel **H** pilih *verydark*.
2. Di bawah variabel **Jenis_kopi** pilih *very_dark*

3. Dalam *field Weight*, set bobot *rule*. Bobot *rule* diset 1 secara *default*.

4. Klik *Add Rule*

Hasil dari langkah-langkah di atas dapat dilihat pada tampilan *rule editor* dalam gambar 4.8 berikut



Gambar 4.8 Pembentukan Rule Melalui Rule Editor

4.2.3.4 Defuzzifikasi

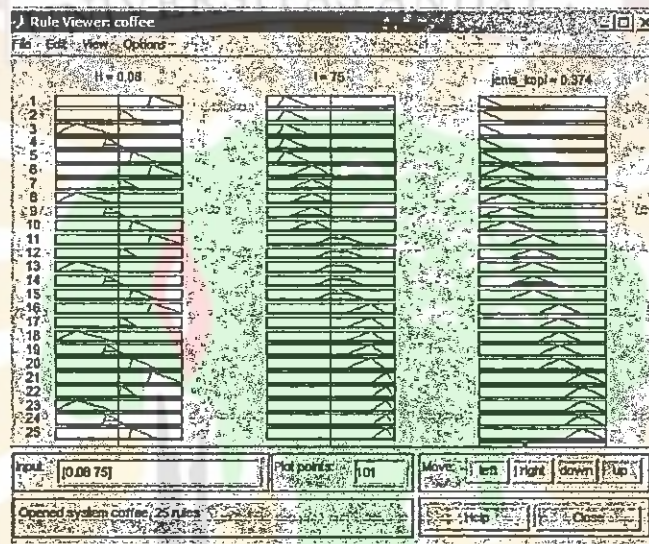
Defuzzifikasi adalah proses pengubahan besaran *Fuzzy* yang disajikan dalam bentuk himpunan-himpunan *Fuzzy* keluaran dengan fungsi keanggotaannya untuk mendapatkan kembali bentuk tegasnya (*crisp*).

Pada tahap defuzzifikasi ini, semua nilai keluaran *Fuzzy* yang dihasilkan pada proses evaluasi *rule* dikombinasikan dengan Fungsi Keanggotaan keluaran untuk mendapatkan keluaran sesuai sistem yang diinginkan.

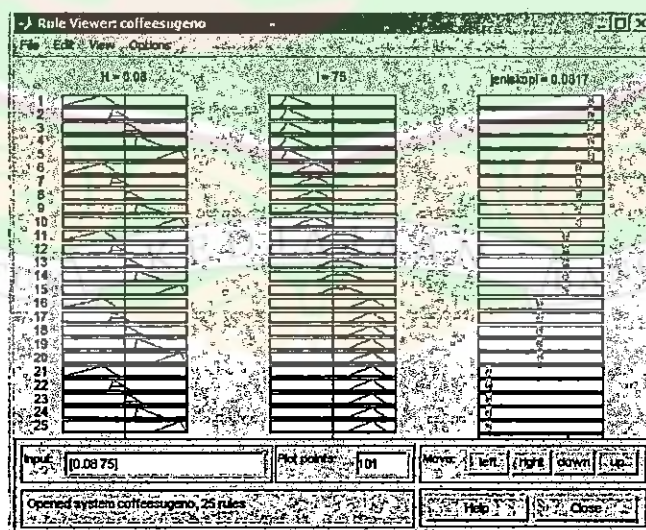
Berdasarkan hasil derajat keanggotaan parameter keluaran (gambar 4.7)

dilakukan proses FIS (*Fuzzy Inference System*) model Mamdani dan Sugeno. Pada FIS Mamdani menggunakan metode *centroid*, sedangkan pada FIS Sugeno menggunakan metode *weighted average*.

Gambar 4.9 dan 4.10 menunjukkan *rule viewer* FIS Mamdani dan FIS Sugeno.



Gambar 4.9 Rule viewer FIS Mamdani



Gambar 4.10 Rule viewer FIS Sugeno

4.2.4 Perancangan *form* GUI Pengaduk Kopi Otomatis

Setelah sistem pengolahan citra dan penarik kesimpulan logika *fuzzy* dirancang untuk menentukan jenis kopi, maka dapat dibuat *software* yang berbentuk *form Graphical User Interface* (GUI) pada MATLAB yang berfungsi sebagai antarmuka bagi *user* dalam menggunakan *software* ini.

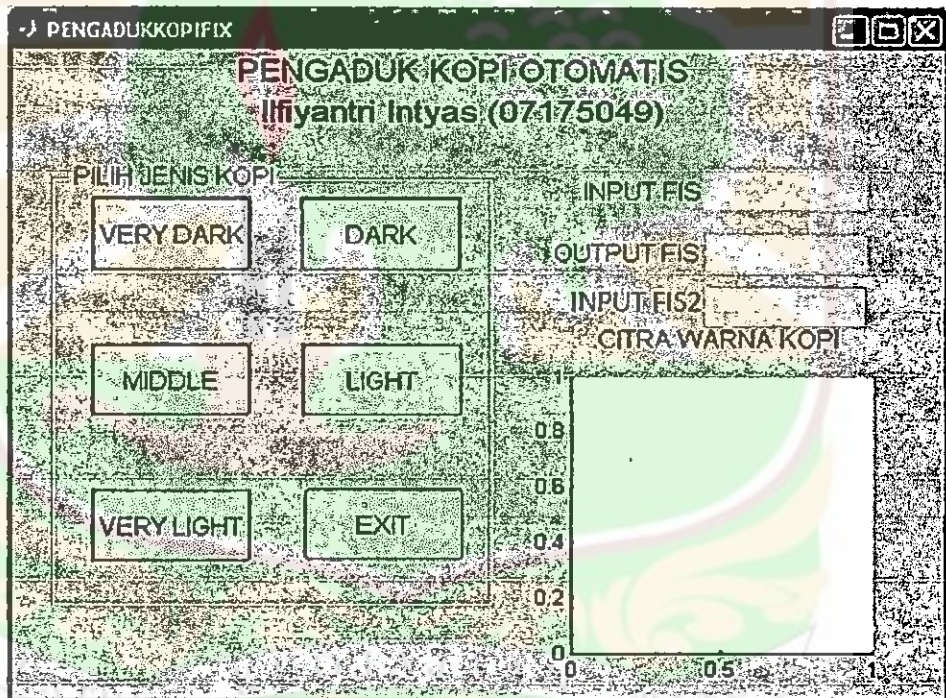
Dalam GUI yang dirancang tersebut terdapat sebuah panel yang berisi enam buah tombol *command* yang masing masing berupa tombol *Very Dark*, *Dark*, *Middle*, *Light*, *Very Light* dan *Exit*, satu buah grafik yang merupakan tempat menampilkan visualisasi dari citra kopi yang ditangkap oleh kamera, lalu dua *static text* yang masing-masing terdapat satu buah kotak *static text* yang merupakan nilai *output* FIS dan *input* FIS.

1. Tombol-tombol yang masing-masing bertuliskan satu jenis kopi, berfungsi untuk menjalankan program. Motor DC dan/atau motor stepper akan bergerak, dan *webcam* yang diletakkan tepat di depan kamera akan langsung diaktifkan oleh program untuk menampilkan citra video kopi, dan meng-*capture frame* citra. Kemudian nilai RGB citra yang di-*capture* oleh kamera akan diolah oleh program untuk mendapatkan nilai *hue* dari citra yang dibutuhkan untuk input logika *fuzzy*. Setelah dilakukan proses FIS maka didapatkan keputusan berupa jenis kopi. Apabila jenis kopi yang terdeteksi telah sesuai dengan jenis kopi pada tombol yang ditekan di awal proses maka komputer akan mengirim data ke mikroprosesor melalui port paralel. Data yang diterima mikrokontroler dari *memory buffer* akan digunakan

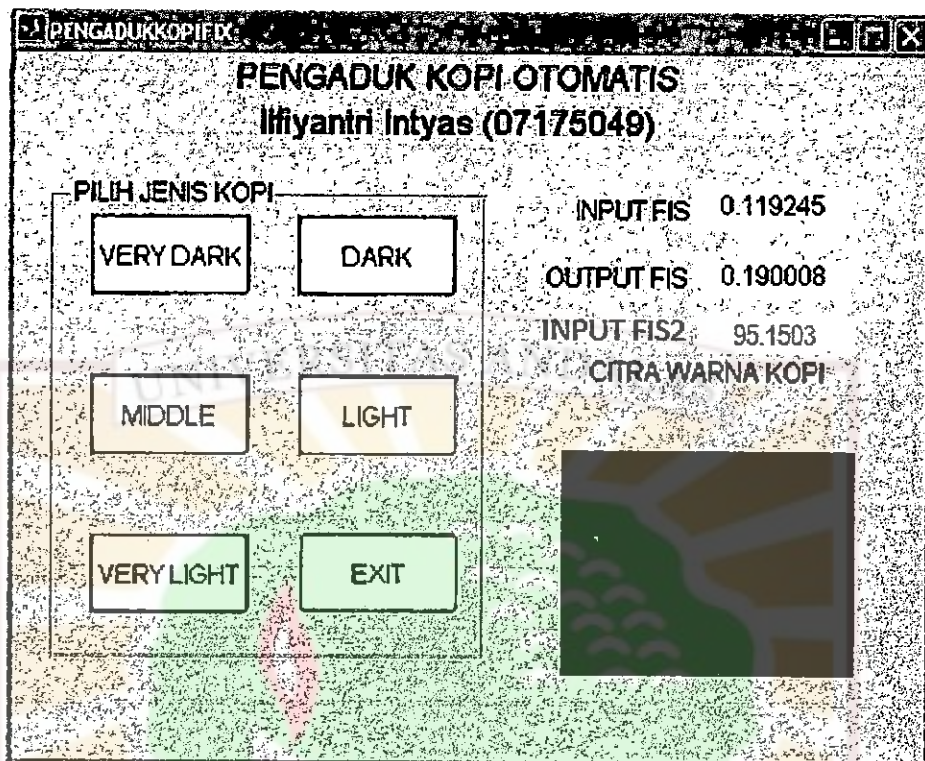
mikrokontroler untuk menghentikan pergerakan motor DC dan motor stepper. Apabila jenis kopi yang terdeteksi belum sesuai dengan jenis kopi yang diminta di awal proses maka program akan kembali meng-*capture* citra untuk memproses *frame* selanjutnya.

2. Tombol *Exit*, berfungsi untuk keluar dari tampilan GUI.

Pada gambar 4.11 berikut adalah tampilan *form* GUI program Pengaduk Kopi Otomatis yang belum dijalankan, dan gambar 4.12 adalah tampilan *form* GUI Pengaduk Kopi Otomatis yang telah dijalankan.



Gambar 4.11 Tampilan *form* GUI Sebelum Dijalankan



Gambar 4.12 Tampilan form GUI Setelah Dijalankan

4.3 I/O Interface

4.3.1 Port Paralel

Pada penelitian tugas akhir ini, *port* paralel digunakan untuk komunikasi komputer PC dengan mikrokontroler pada alat pengaduk kopi. Keluaran dari *port* paralel ini akan menjadi masukan bagi mikrokontroler yaitu pada *port* A sehingga mikrokontroler dapat mengatur motor DC sebagai pengaduk kopi dan motor stepper sebagai penggerak katup corong krimer sesuai dengan jenis kopi yang diinginkan. Pin pada port paralel yang digunakan hanya Port Data (DP0-DP7) sehingga komputer PC hanya memberikan keluaran pada *port* paralel dan tidak ada komunikasi 2 arah di antara peralatan ini

4.3.2 Mikrokontroller ATMEGA 8535

Mikrokontroller ini digunakan untuk mengendalikan motor DC sebagai pengaduk kopi dan motor stepper sebagai pengatur katup corong krim. Mikrokontroller ini terdiri dari 40 pin, 32 pin merupakan input/output. Input/output mikrokontroller ini adalah port A, port B, port C dan port D. Pada penelitian ini, port A digunakan sebagai input dari port paralel, port B.1 sebagai output pada motor DC dan port C.0 - C.3 sebagai output pada motor stepper, sedangkan port D tidak digunakan.

Algoritma pada mikrokontroller adalah :

1. Baca data input pada port A.
2. Jika data input bernilai 0xFE (254), maka mikrokontroller akan mengaktifkan motor DC sebagai pengaduk kopi.
3. Jika data input bernilai 0xFD (253), maka mikrokontroller akan mengaktifkan motor DC sebagai pengaduk kopi dan motor stepper sebagai penggerak katup corong krim. Pada kondisi ini motor stepper akan bergerak cukup cepat.
4. Jika data input bernilai 0xFC (252), maka mikrokontroller akan mengaktifkan motor DC sebagai pengaduk kopi dan motor stepper sebagai penggerak katup corong krim. Pada kondisi ini motor stepper bergerak lebih lama agar krim yang turun lebih banyak.
5. Jika data input bernilai 0xFB (251), maka mikrokontroller akan menghentikan semua motor yang bergerak.

4.3.3 Motor DC

Pada penelitian ini, motor DC digunakan sebagai pengaduk kopi. Untuk menggerakkan motor DC diperlukan *driver* motor DC yang terdiri dari sebuah resistor, transistor dan dioda.

Algoritma pada motor DC adalah :

1. Jika data masukan port A mikrokontroller bernilai 0xFE, 0xFD dan 0xFC, maka motor DC akan aktif (port B.1 berlogika 1).
2. Jika data masukan port A mikrokontroller bernilai 0xFB maka motor DC akan berhenti (port B.1 berlogika 0).

4.3.4 Motor Stepper

Motor stepper digunakan sebagai pengatur katup corong krimer. Besarnya bukaan katup corong krimer sangat bergantung pada data yang dikirim mikrokontroler pada port C.0 – C.3. Untuk menggerakkan motor stepper juga diperlukan *driver* motor stepper.

Algoritma pada motor stepper adalah:

1. Jika data masukan port A bernilai 0xFE, motor stepper diam (tidak melakukan gerakan apapun).
2. Jika data masukan port A bernilai 0xFD, motor stepper akan bergerak cukup cepat (delay yang diberikan lebih kecil).
3. Jika data masukan port A bernilai 0xFC, motor stepper akan bergerak lebih lama (delay yang diberikan lebih besar).

Jika data masukan port A bernilai 0xFB, motor stepper akan berhenti bergerak.

4.4 Evaluasi Terhadap Hasil Penelitian

Evaluasi terhadap hasil penelitian merupakan tolak ukur dari keberhasilan penelitian ini serta dapat dijadikan sebagai dasar bagi pengembangan sistem lebih lanjut. Evaluasi hasil juga merupakan bagian dari teknik analisa sistem pada tugas akhir ini. Evaluasi hasil yang dilakukan adalah membandingkan apakah hasil yang didapat melalui eksperimen sama dengan keadaan yang sebenarnya.



BAB V

ANALISA DAN HASIL PEMBAHASAN

Hasil dari keluaran sistem pada penelitian ini merupakan pergerakan pengaduk kopi dan penutup saluran krimer secara otomatis berdasarkan warna citra kopi yang divisualisasikan secara langsung berupa gerakan memutar pengaduk kopi serta visualisasi melalui GUI MATLAB. Adapun analisa terhadap hasil dilakukan secara objektif dengan cara membandingkan secara langsung gerak yang dilakukan pengaduk kopi dan penutup saluran krimer berdasarkan warna citra kopi yang terdeteksi oleh sistem dengan gerak yang seharusnya dilakukan pengaduk kopi dan penutup saluran krimer berdasarkan warna citra kopi yang diamati dengan cara observasi warna kopi secara langsung.

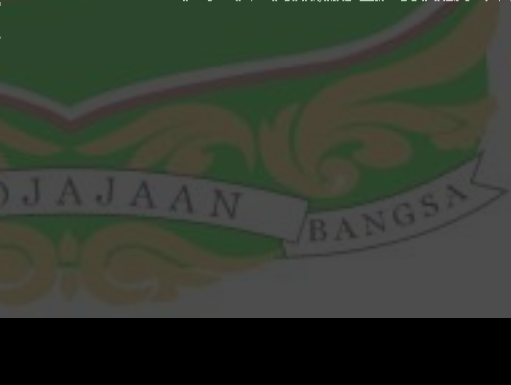
Hasil-hasil yang didapatkan melalui eksperimen (*output* sistem) akan dibandingkan dengan *input* sistem (keadaan sebenarnya). Dalam penelitian ini digunakan 5 macam jenis kopi, yaitu Sangat Pekat/ *Very Dark*, Pekat/*Dark*, Biasa/*Middle*, dan Ringan/ *Light*, dan Sangat Ringan/*Very Light*. Hasil dari proses otomatisasi sistem pergerakan pengaduk kopi yang akan dibahas pada penelitian ini difokuskan pada analisa hasil *capture* citra, dan penarikan kesimpulan FIS (*Fuzzy Inference System*) tipe Mamdani dan Sugeno.

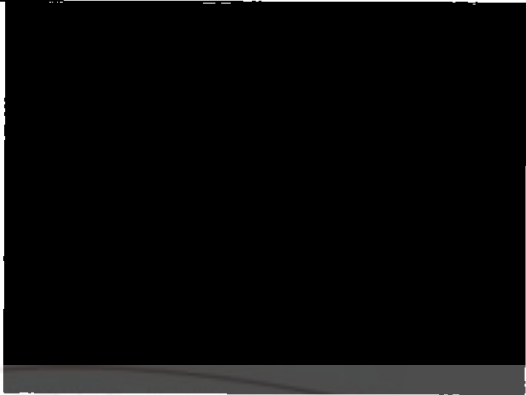
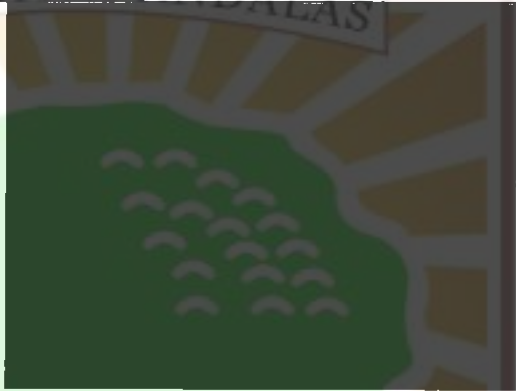
5.1 Hasil *capture*

Sampel citra yang digunakan adalah sampel citra kopi yang di-*capture* oleh kamera. Citra tersebut dapat dilihat pada tabel 5.1. Pengambilan sampel ini dibagi menjadi lima bagian kondisi pengambilan sampel, yaitu kopi Sangat

Pekat/ *Very Dark*, Pekat/*Dark*, Biasa/*Middle*, dan Ringan/ *Light*, dan Sangat Ringan/*Very Light*.

Tabel 5.1 Citra kopi yang ditangkap oleh kamera

NO	Jenis Kopi	Citra Kopi
1	Sangat Pekat/ <i>Very Dark</i>	
2	Pekat/ <i>Dark</i>	
3	Biasa/ <i>Middle</i>	

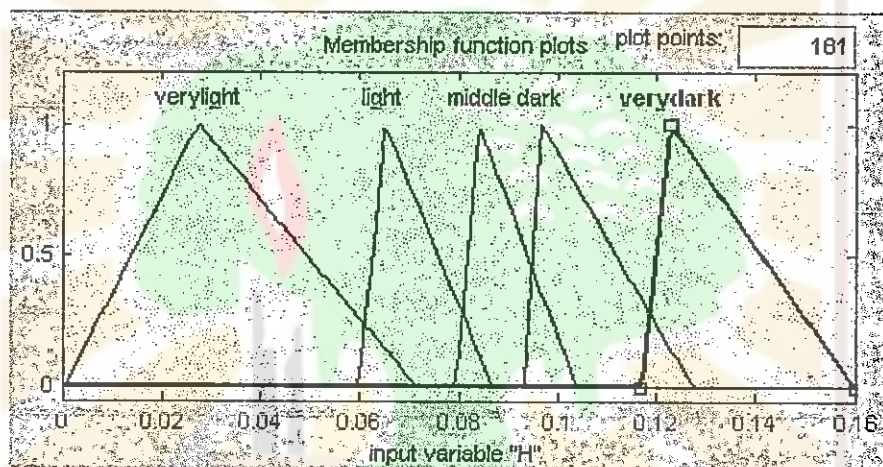
4	Ringan/ <i>Light</i>	
5	Sangat Ringan/ <i>Very Light</i>	

Dari sampel citra hasil pengujian pada jenis kopi Sangat Pekat/ *Very Dark*, Pekat/ *Dark*, Biasa/ *Middle*, dan Ringan/ *Light*, dan Sangat Ringan/ *Very Light* yang ditunjukkan pada tabel 5.1 dapat dilihat bahwa sistem sudah dapat melakukan operasi pengolahan citra untuk jenis kopi Sangat Pekat/ *Very Dark*, Pekat/ *Dark*, Biasa/ *Middle*, dan Ringan/ *Light*, dan Sangat Ringan/ *Very Light*. Pada citra hasil *capture* terdapat sedikit warna putih yang disebabkan oleh pantulan cahaya (kilap) pada gelas kopi. Warna putih ini tidak mempengaruhi sistem FIS, karena citra kopi yang di-*capture* akan melalui proses *cropping* dahulu pada wilayah citra yang tidak tertutup kilap gelas.

5.2 Pengujian *Fuzzy Inference System* dan Kinerja Mesin Pengaduk Kopi.

5.2.1 *Membership Function*

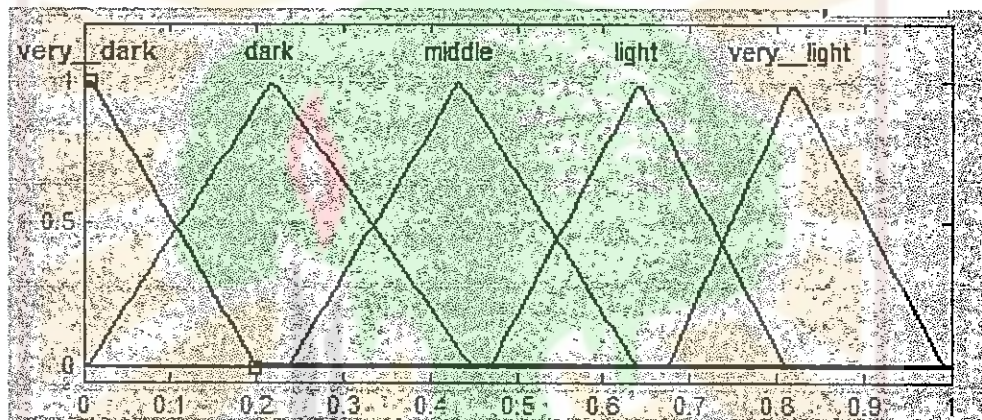
Pembentukan *MembershipFunction* terbagi dua yaitu *Membership Function* masukan dan *Membership Function* keluaran. Untuk *Membership Function* masukan, terdapat dua variabel *input* yaitu variabel *H* dan *I*. Sedangkan untuk variabel keluaran terdapat lima *Membership Function* yaitu jenis kopi *verylight*, *light*, *middle*, *dark* dan *very dark*. Untuk *Membership Function* masukan dapat dilihat pada Gambar 5.1 dan *Membership Function* keluaran pada Gambar 5.2 dan Gambar 5.3



Gambar 5.1 Variabel Masukan *H*

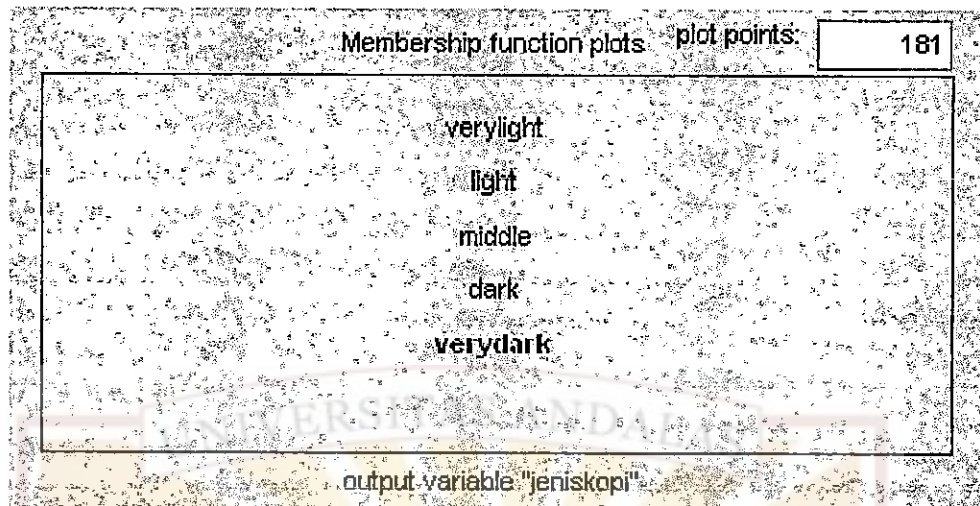
Pada variabel *input H*, rentang semesta pembicaraan adalah dari 0 – 0.16. Pada *Membership Function verylight*, memiliki parameter [0.0005 0.0273 0.0717] dengan artian derajat keanggotaan 1 pada 0.0273 dan sisi miring pada rentang 0.0005 – 0.0273 dan 0.0273 – 0.0717. Pada *MembershipFunctionlight*, memiliki parameter [0.0599 0.06497 0.087] dengan artian derajat keanggotaan 1 pada nilai 0.06497, sisi miring sebelah kiri pada rentang 0.0599 - 0.06497 dan sisi miring sebelah kanan pada rentang 0.06497-0.087. Pada *Membership Function middle*, parameternya [0.0794 0.08402 0.104] dengan artian derajat keanggotaan 1 pada nilai 0.08402, sisi miring sebelah kiri pada rentang 0.0794-

0.08402 dan sisi miring sebelah kanan pada rentang 0.08402-0.104. Pada *Membership Function dark*, parameternya [0.0937 0.0971 0.128] dengan artian derajat keanggotaan 1 pada nilai 0.0971, sisi miring sebelah kiri pada rentang 0.0933-0.0971 dan sisi miring sebelah kanan pada rentang 0.0971-0.128. Pada *Membership Function verydark*, parameternya [0.117 0.123 0.16] dengan artian derajat keanggotaan 1 pada nilai 0.123, sisi miring sebelah kiri pada rentang 0.117-0.123 dan sisi miring sebelah kanan pada rentang 0.13-0.16.



Gambar 5.2 Variabel *Output FIS Mamdani*

Untuk variabel keluaran ini, terdapat lima *Membership Function* dengan label berupa jenis kopi. Pada *Membership Function very_dark*, memiliki parameter [-0.3914 0.008236 0.1996]. Pada *Membership Function dark*, memiliki parameter [0.00401 0.2183 0.448]. Pada *Membership Function middle*, memiliki [0.2369 0.4339 0.6389]. Pada *Membership Function light*, memiliki parameter [0.47 0.6415 0.811]. Pada *Membership Function very_light*, memiliki parameter [0.6741 0.8191 0.9931].



Gambar 5.3 Variabel *Output* FIS Sugeno

Untuk variable keluaran pada FIS Sugeno, terdapat lima *MembershipFunction* dengan label berupa jenis kopi yang dikenali. Parameter untuk masing-masing *MembershipFunction* tersebut adalah *very dark* = 0.1, *dark* = 0.09, *middle* = 0.08, *light* = 0.06, dan *very light* = 0.02.

5.2.2 Rule yang Digunakan.

Untuk dapat mengenali jenis kopi dalam sistem, maka pada inferensi *fuzzy* ini dibutuhkan suatu aturan (*rule*) yang akan digunakan pada setiap jenis kopi. *Rule* ini didapatkan dengan melihat data-data hasil peningkatan kualitas citra dari setiap jenis kopi yang diujikan dan membandingkan data-data tersebut dengan *MembershipFunction* masukan yang telah dibentuk sebelumnya. Misalnya untuk jenis kopi *very dark* didapatkan nilai *hue* 0.119245. Data yang didapat tersebut dibandingkan dengan kelima *MembershipFunction* masukan dengan cara melihat nilai pada variabel apakah nilai tersebut berada pada area *verylight*, *light*, *middle*, *dark* atau *verydark* dalam *membership function* masukan. Pada data yang didapatkan di

atas, maka *rule* yang akan terbentuk dari variabel-variabel yang ada adalah *verydark*.

Untuk melihat semua *rule* yang terbentuk untuk jenis kopi Sangat Pekat/*Very Dark*, Pekat/*Dark*, Biasa/*Middle*, dan Ringan/*Light*, dan Sangat Ringan/*Very Light* dapat dilihat pada Tabel 5.2 berikut.

Tabel 5.2 *Rule* yang digunakan pada FIS

H \ I	<i>Very Low</i>	<i>Low</i>	<i>Middle</i>	<i>High</i>	<i>Very High</i>
<i>Very Low</i>	<i>Very Dark</i>	<i>Dark</i>	<i>Middle</i>	<i>Light</i>	<i>Very Light</i>
<i>Low</i>	<i>Very Dark</i>	<i>Dark</i>	<i>Middle</i>	<i>Light</i>	<i>Very Light</i>
<i>Middle</i>	<i>Very Dark</i>	<i>Dark</i>	<i>Middle</i>	<i>Light</i>	<i>Very Light</i>
<i>High</i>	<i>Very Dark</i>	<i>Dark</i>	<i>Middle</i>	<i>Light</i>	<i>Very Light</i>
<i>Very High</i>	<i>Very Dark</i>	<i>Dark</i>	<i>Middle</i>	<i>Light</i>	<i>Very Light</i>

5.2.3 Hasil Pengujian Kopi Dengan Mesin Kopi Otomatis

Sistem pengaduk kopi otomatis ini dirancang dalam bentuk gabungan *software* dan *hardware*. *Software* yang dirancang pada penelitian ini merupakan program yang berfungsi sebagai pengenalan warna dan *Fuzzy Inference System*.

5.2.3.1 Hasil Pengujian dengan menggunakan FIS Mamdani

Tabel 5.3 sampai tabel 5.7 menunjukkan hasil pengujian kopi menggunakan mesin pengaduk kopi berdasarkan lima jenis kopi yang di ujikan pada sistem ini serta tingkat keberhasilannya. Sampel yang ditunjukkan pada

tabel adalah 20 kali pengujian secara berulang untuk masing-masing kondisi tingkat keburaman berdasarkan observasi visual secara langsung.

Pada tabel juga ditunjukkan nilai *output crisp* yang diperoleh sistem dari jenis kopi yang diobservasi. Sedangkan pada sistem yang dirancang, rentang nilai *output crisp* yang digunakan adalah; 0.7 sampai 0.9 untuk jenis kopi *very light*, besar dari 0.55 sampai 0.7 untuk jenis kopi *light*, besar dari 0.3 sampai 0.55 untuk jenis kopi *middle*, besar dari 0.183-0.3 untuk jenis kopi *dark*, dan nilai *output crisp* yang lebih kecil dari 0.183 untuk jenis kopi *very dark*.

Tabel 5.3 Hasil Pengujian Kopi untuk Jenis Kopi *very dark*

No	Jenis Kopi	Output Crisp	Kebenaran Jenis Kopi	Keterangan
1	<i>Very dark</i>	0.121892	BENAR	-
2	<i>Very dark</i>	0.131382	BENAR	-
3	<i>Very dark</i>	0.082982	BENAR	-
4	<i>Very dark</i>	0.121655	BENAR	-
5	<i>Very dark</i>	0.098336	BENAR	-
6	<i>Very dark</i>	0.124531	BENAR	-
7	<i>Very dark</i>	0.095561	BENAR	-
8	<i>Very dark</i>	0.143100	BENAR	-
9	<i>Very dark</i>	0.078000	BENAR	-
10	<i>Very dark</i>	0.076400	BENAR	-
11	<i>Very dark</i>	0.072900	BENAR	-
12	<i>Very dark</i>	0.076900	BENAR	-

13	<i>Very dark</i>	0.080300	BENAR	-
14	<i>Very dark</i>	0.065900	BENAR	-
15	<i>Very dark</i>	0.071600	BENAR	-
16	<i>Very dark</i>	0.076300	BENAR	-
17	<i>Very dark</i>	0.068100	BENAR	-
18	<i>Very dark</i>	0.072800	BENAR	-
19	<i>Very dark</i>	0.069300	BENAR	-
20	<i>Very dark</i>	0.077800	BENAR	-
Persentase Kebenaran Jenis Kopi			100%	
Persentase Kesalahan Jenis Kopi			0%	

Dari tabel-tabel pengujian 5.3 di atas terlihat bahwa pada pengujian jenis kopi Sangat Pekat/ *Very Dark* , tingkat keberhasilan pengujian mencapai 100 persen.

Tabel 5.4 Hasil Pengujian Kopi untuk Jenis Kopi *dark*

No	Jenis Kopi	Output Crisp	Kebenaran Jenis Kopi	Keterangan
1	<i>Dark</i>	0.224702	BENAR	-
2	<i>Dark</i>	0.224240	BENAR	-
3	<i>Dark</i>	0.223715	BENAR	-
4	<i>Dark</i>	0.210603	BENAR	-
5	<i>Dark</i>	0.224564	BENAR	-
6	<i>Dark</i>	0.223889	BENAR	-
7	<i>Dark</i>	0.224554	BENAR	-

8	Dark	-	SALAH	Motor terus berputar sampai tak terbatas sehingga <i>output</i> di luar jangkauan <i>range output crisp dark</i> .
9	Dark	0.224223	BENAR	-
10	Dark	0.223696	BENAR	-
11	Dark	0.240693	BENAR	-
12	Dark	0.224379	BENAR	-
13	Dark	0.223728	BENAR	-
14	Dark	-	SALAH	Motor terus berputar sampai tak terbatas sehingga <i>output</i> di luar jangkauan <i>range output crisp dark</i> .
15	Dark	0.224562	BENAR	-
16	Dark	0.224295	BENAR	-
17	Dark	0.219786	BENAR	-
18	Dark	0.223966	BENAR	-
19	Dark	0.190008	BENAR	-
20	Dark	0.274201	BENAR	-
Persentase Kebenaran Jenis Kopi			18/10 X 100% = 90%	
Persentase Kesalahan Jenis Kopi			2/10 X 100% = 10%	

Sedangkan pada tabel 5.4 di atas ditunjukkan bahwa untuk jenis kopi *dark*, masih terdapat 2 kali kesalahan pengenalan warna dari 20 kali pengujian. Kesalahan ini dapat terjadi karena nilai *inputFIS* berada di luar rentang yang

diujikan, sehingga *output crisp* yang dihasilkan tidak lagi berada pada rentang kopi *dark*. Kesalahan ini menyebabkan pengaduk kopi akan terus mengaduk kopi, dan penutup saluran krimer akan terus membuka dan menutup sehingga krimer akan terus ditambahkan ke dalam kopi.

Tabel 5.5 Hasil Pengujian Kopi untuk Jenis Kopi *middle*

No	Jenis Kopi	Output Crisp	Kebenaran Jenis Kopi	Keterangan
1	Middle	0.436647	BENAR	-
2	Middle	0.436595	BENAR	-
3	Middle	0.461254	BENAR	-
4	Middle	0.511304	BENAR	-
5	Middle	0.509616	BENAR	-
6	Middle	0.445301	BENAR	-
7	Middle	0.461942	BENAR	-
8	Middle	0.513809	BENAR	-
9	Middle	0.5	SALAH	Pada <i>toolbox</i> Fuzzy matlab, jika <i>input</i> FIS berada di luar <i>range</i> yang disediakan, <i>output</i> FIS akan bernilai 0.5.
10	Middle	0.309233	BENAR	-
11	Middle	0.356048	BENAR	-
12	Middle	0.452766	BENAR	-
13	Middle	0.436791	BENAR	-
14	Middle	0.436847	BENAR	-

15	Middle	0.436828	BENAR	-
16	Middle	0.5	SALAH	Pada <i>toolbox</i> Fuzzy matlab, jika <i>input</i> FIS berada di luar <i>range</i> yang disediakan, <i>output</i> FIS akan bernilai 0.5.
17	Middle	0.471754	BENAR	-
18	Middle	0.436811	BENAR	-
19	Middle	0.353097	BENAR	-
20	Middle	0.309329	BENAR	-
Persentase Kebenaran			18/20 X 100% = 90%	
Jenis Kopi				
Persentase Kesalahan			2/20 X 100% = 10%	
Jenis Kopi				

Pada tabel 5.5 di atas ditunjukkan bahwa untuk jenis kopi *middle*, masih terdapat 2 kali kesalahan pengenalan warna dari 20 kali pengujian. Kesalahan ini dapat terjadi karena nilai *input*FIS berada di luar rentang semesta pembicaraan, sehingga *output crisp* yang dihasilkan adalah 0.5. Walaupun nilai *outout crisp* yang dihasilkan masih dalam rentang yang ditentukan untuk *output*jenis kopi *dark*, namun hasil ini dianggap salah karena *input*FIS yang didapatkan berada di luar rentang semesta pembicaraan.

Tabel 5.6 Hasil Pengujian Kopi untuk Jenis Kopi *light*

No	Jenis Kopi	Output Crisp	Kebenaran Jenis Kopi	Keterangan
1	Light	0.660427	BENAR	-

2	<i>Light</i>	0.661190	BENAR	-
3	<i>Light</i>	0.660827	BENAR	-
4	<i>Light</i>	0.653107	BENAR	-
5	<i>Light</i>	0.679315	BENAR	-
6	<i>Light</i>	0.661737	BENAR	-
7	<i>Light</i>	0.6555365	BENAR	-
8	<i>Light</i>	0.6617370	BENAR	-
9	<i>Light</i>	0.6406580	BENAR	-
10	<i>Light</i>	0.664174	BENAR	-
11	<i>Light</i>	0.640764	BENAR	-
12	<i>Light</i>	0.640732	BENAR	-
13	<i>Light</i>	-	SALAH	Motor terus berputar sampai tak terbatas sehingga <i>output</i> di luar jangkauan <i>range output crisp light</i> .
14	<i>Light</i>	-	SALAH	Motor terus berputar sampai tak terbatas sehingga <i>output</i> di luar jangkauan <i>range output crisp light</i> .
15	<i>Light</i>	0.692046	BENAR	-
16	<i>Light</i>	0.64079	BENAR	-
17	<i>Light</i>	-	SALAH	Motor terus berputar sampai tak terbatas sehingga <i>output</i> di luar jangkauan <i>range output crisp light</i> .

18	<i>Light</i>	0.640791	BENAR	-
19	<i>Light</i>	0.640783	BENAR	-
20	<i>Light</i>	0.640787	BENAR	-
Persentase Kebenaran Jenis Kopi		$17/20 \times 100\% = 85\%$		
Persentase Kesalahan Jenis Kopi		$3/20 \times 100\% = 15\%$		

Pada tabel 5.6 di atas ditunjukkan bahwa untuk jenis kopi *light*, masih terdapat 3 kali kesalahan pengenalan warna dari 20 kali pengujian. Kesalahan ini dapat terjadi karena nilai *inputFIS* berada di luar rentang semesta pembicaraan, sehingga *output crisp* yang dihasilkan tidak lagi berada pada rentang kopi *light*. Kesalahan ini menyebabkan pengaduk kopi akan terus mengaduk kopi, dan penutup saluran krimer akan terus membuka dan menutup sehingga krimer akan terus ditambahkan ke dalam kopi.

Tabel 5.7Hasil Pengujian Kopi untuk Jenis Kopi *verylight*.

No	Jenis Kopi	<i>Output Crisp</i>	Kebenaran Jenis Kopi	Keterangan
1	<i>Very Light</i>	0.78347	BENAR	-
2	<i>Very Light</i>	0.829493	BENAR	-
3	<i>Very Light</i>	0.799121	BENAR	-
4	<i>Very Light</i>	-	SALAH	Motor terus berputar sampai tak terbatas sehingga <i>output</i> di luar jangkauan <i>range output crisp very light</i> .
5	<i>Very Light</i>	0.829645	BENAR	-
6	<i>Very Light</i>	0.829869	BENAR	-

7	<i>Very Light</i>	0.81715	BENAR	-
8	<i>Very Light</i>	0.830355	BENAR	-
9	<i>Very Light</i>	0.828828	BENAR	-
10	<i>Very Light</i>	0.832449	BENAR	-
11	<i>Very Light</i>	0.832449	BENAR	-
12	<i>Very Light</i>	0.830357	BENAR	-
13	<i>Very Light</i>	0.830877	BENAR	-
14	<i>Very Light</i>	0.829201	BENAR	-
15	<i>Very Light</i>	0.83093	BENAR	-
16	<i>Very Light</i>	0.740211	BENAR	-
17	<i>Very Light</i>	0.64079	BENAR	-
18	<i>Very Light</i>	0.64076	BENAR	-
19	<i>Very Light</i>	0.801257	BENAR	-
20	<i>Very Light</i>	0.829498	BENAR	-
Persentase Kebenaran Jenis Kopi			$19/20 \times 100\% = 95\%$	
Persentase Kesalahan Jenis Kopi			$1/20 \times 100\% = 5\%$	

Pada tabel 5.7 di atas ditunjukkan bahwa untuk jenis kopi *very light*, masih terdapat 1 kali kesalahan pengenalan warna dari 20 kali pengujian. Kesalahan ini dapat terjadi karena nilai *inputFIS* berada di luar rentang yang diujikan, sehingga *output crisp* yang dihasilkan adalah 0.5 yang mana berada di luar *rangeoutputlight*. Kesalahan ini menyebabkan pengaduk kopi akan terus mengaduk kopi, dan penutup saluran krimer akan terus membuka dan menutup sehingga krimer akan terus ditambahkan ke dalam kopi.

Tabel 5.8 Performansi total mesin pengaduk kopi otomatis dengan FIS

Mamdani.

No	Jenis Kopi	Persentase Kebenaran Pengujian	Persentase Kesalahan Pengujian
1	<i>Very dark</i>	100%	0%
2	<i>Dark</i>	90%	10%
3	<i>Middle</i>	90%	10%
4	<i>Light</i>	85%	15%
5	<i>Very light</i>	95%	5%
Persentase Rata-rata		$460/500 \times 100\% = 92\%$	$40/500 \times 100\% = 8\%$

Dari tabel 5.8 di atas ditunjukkan persentase rata-rata keberhasilan adalah sebesar 92% yang secara keseluruhan sudah baik, sedangkan persentase rata-rata kesalahan adalah sebesar 8%. Kesalahan terbanyak terjadi pada jenis kopi Ringan/ *light* yaitu sebanyak 3 kali dari 20 kali pengujian. Total kesalahan terjadi sebanyak 8 kali dari 100 pengujian. Dari kesalahan-kesalahan yang terjadi dapat disimpulkan bahwa hal ini dapat terjadi karena nilai *input* FIS yang beradadi luar rentang semesta pembicaraan.

5.2.3.2 Hasil pengujian dengan FIS Sugeno

Tabel 5.9 sampai tabel 5.13 menunjukkan hasil pengujian kopi menggunakan mesin pengaduk kopi berdasarkan lima jenis kopi yang diujikan pada sistem ini serta tingkat keberhasilannya. Sampel yang ditunjukkan pada tabel

adalah 20 kali pengujian secara berulang untuk masing-masing jenis kopi berdasarkan observasi visual secara langsung.

Pada tabel juga ditunjukkan nilai *output crisp* yang diperoleh sistem dari kondisi jenis kopi yang diobservasi. Untuk variabel keluaran, terdapat lima *MembershipFunction* dengan label berupa jenis kopi yang dikenali. Parameter untuk masing-masing *MembershipFunction* tersebut adalah *very dark* = 0.1, *dark* = 0.09, *middle* = 0.08, *light* = 0.06, dan *very light* = 0.02.

Tabel 5.9 Pengujian Kopi untuk Jenis Kopi *very dark*

No	Jenis Kopi	Output Crisp	Kebenaran Jenis Kopi	Keterangan
1	<i>Very dark</i>	0.1	BENAR	-
2	<i>Very dark</i>	0.1	BENAR	-
3	<i>Very dark</i>	0.1	BENAR	-
4	<i>Very dark</i>	0.1	BENAR	-
5	<i>Very dark</i>	-	SALAH	Motor terus berputar
6	<i>Very dark</i>	0.1	BENAR	-
7	<i>Very dark</i>	0.1	BENAR	-
8	<i>Very dark</i>	0.1	BENAR	-
9	<i>Very dark</i>	0.099825	BENAR	-
10	<i>Very dark</i>	0.1	BENAR	-
11	<i>Very dark</i>	0.1	BENAR	-
12	<i>Very dark</i>	0.0987745	BENAR	-
13	<i>Very dark</i>	0.0974443	BENAR	-
14	<i>Very dark</i>	0.0991059	BENAR	-

15	<i>Very dark</i>	-	SALAH	Motor terus berputar
16	<i>Very dark</i>	0.0717975	BENAR	-
17	<i>Very dark</i>	0.0991272	BENAR	-
18	<i>Very dark</i>	0.0990014	BENAR	-
19	<i>Very dark</i>	0.1	BENAR	-
20	<i>Very dark</i>	0.1	BENAR	-
Persentase Kebenaran Jenis Kopi			18/20 X 100% = 90%	
Persentase Kesalahan Jenis Kopi			2/20 X 100% = 10%	

Pada tabel 5.9 di atas ditunjukkan bahwa untuk jenis kopi *dark*, masih terdapat 2 kali kesalahan pengenalan warna dari 20 kali pengujian. Kesalahan ini dapat terjadi karena nilai *input* FIS berada di luar rentang semesta pembicaraan, sehingga *output crisp* yang dihasilkan tidak lagi berada pada rentang *output dark*. Kesalahan ini menyebabkan pengaduk kopi akan terus mengaduk kopi.

Tabel 5.10 Pengujian Kopi untuk Jenis Kopi *dark*

No	Jenis Kopi	<i>Output Crisp</i>	Kebenaran Jenis Kopi	Keterangan
1	<i>Dark</i>	0.09	BENAR	-
2	<i>Dark</i>	0.09	BENAR	-
3	<i>Dark</i>	0.09	BENAR	-
4	<i>Dark</i>	0.09	BENAR	-
5	<i>Dark</i>	0.089571	BENAR	-
6	<i>Dark</i>	0.09	BENAR	-
7	<i>Dark</i>	0.09	BENAR	-

8	<i>Dark</i>	0.0884114	BENAR	-
9	<i>Dark</i>	0.0897199	BENAR	-
10	<i>Dark</i>	0.088288	BENAR	-
11	<i>Dark</i>	0.0891854	BENAR	-
12	<i>Dark</i>	0.0894141	BENAR	-
13	<i>Dark</i>	0.0884374	BENAR	-
14	<i>Dark</i>	0.09	BENAR	-
15	<i>Dark</i>	0.0879309	BENAR	-
16	<i>Dark</i>	0.09	BENAR	-
17	<i>Dark</i>	0.09	BENAR	-
18	<i>Dark</i>	0.09	BENAR	-
19	<i>Dark</i>	0.0882943	BENAR	-
20	<i>Dark</i>	0.0880481	BENAR	-
Persentase Kebenaran Jenis Kopi			100%	
Persentase Kesalahan Jenis Kopi			0%	

Dari tabel-tabel pengujian 5.10 di atas terlihat bahwa pada pengujian jenis kopi Pekat/ *Dark* tingkat keberhasilan pengujian mencapai 100 persen.

Tabel 5.11 Pengujian Kopi untuk Jenis Kopi *middle*.

No	Jenis Kopi	Output Crisp	Kebenaran Jenis Kopi	Keterangan
1	<i>Middle</i>	0.0806254	BENAR	-
2	<i>Middle</i>	0.08	BENAR	-
3	<i>Middle</i>	0.496566	BENAR	-

4	Middle	0.08	BENAR	-
5	Middle	0.08	BENAR	-
6	Middle	0.0797654	BENAR	-
7	Middle	0.08	BENAR	-
8	Middle	0.0780631	BENAR	-
9	Middle	0.0792832	BENAR	-
10	Middle	0.0790557	BENAR	-
11	Middle	0.08	BENAR	-
12	Middle	0.0785252	BENAR	-
13	Middle	0.08	BENAR	-
14	Middle	0.08	BENAR	-
15	Middle	0.0795018	BENAR	-
16	Middle	0.079192	BENAR	-
17	Middle	0.0753403	BENAR	-
18	Middle	0.0737939	BENAR	-
19	Middle	0.0777278	BENAR	-
20	Middle	0.0768478	BENAR	-
Persentase Kebenaran Jenis Kopi			100%	
Persentase Kesalahan Jenis Kopi			0%	

Dari tabel-tabel pengujian 5.11 di atas terlihat bahwa pada pengujian jenis kopi Biasa/ *Middle* , tingkat keberhasilan pengujian mencapai 100 persen.

Tabel 5.12 Pengujian Kopi untuk Jenis *Kopilight*.

No	Jenis	Output	Kebenaran	Keterangan
----	-------	--------	-----------	------------

	Kopi	Crisp	Jenis Kopi	
1	Light	0.658744	BENAR	-
2	Light	0.640822	BENAR	-
3	Light	0.672092	BENAR	-
4	Light	0.0561921	BENAR	-
5	Light	0.689678	BENAR	-
6	Light	-	SALAH	Motor terus berputar sampai tak terbatas sehingga <i>output</i> di luar <i>range output crisp light</i> .
7	Light	0.690258	BENAR	-
8	Light	0.640772	BENAR	-
9	Light	-	SALAH	Motor terus berputar sampai tak terbatas sehingga <i>output</i> di luar <i>range output crisp light</i> .
10	Light	0.640788	BENAR	-
11	Light	0.640728	BENAR	-
12	Light	0.640735	BENAR	-
13	Light	0.640684	BENAR	-
14	Light	0.64081	BENAR	-
15	Light	0.640796	BENAR	-
16	Light	0.655359	BENAR	-
17	Light	0.06	BENAR	-
18	Light	0.06	BENAR	-
19	Light	0.0491694	BENAR	-
20	Light	-	SALAH	Motor terus berputar sampai tak terbatas sehingga <i>output</i> di luar <i>range output crisp light</i> .
Persentase Kebenaran			$17/20 \times 100\% = 85\%$	

Jenis Kopi	
Persentase Kesalahan Jenis Kopi	$3/20 \times 100\% = 15\%$

Pada tabel 5.12 di atas ditunjukkan bahwa untuk jenis kopi *light*, masih terdapat 3 kali kesalahan pengenalan warna dari 20 kali pengujian. Kesalahan ini dapat terjadi karena nilai *inputFIS* berada di luar rentang semesta pembicaraan, sehingga *output crisp* yang dihasilkan tidak lagi berada pada rentang *output light*. Kesalahan ini menyebabkan pengaduk kopi akan terus mengaduk kopi, dan penutup saluran krimer akan terus membuka dan menutup sehingga krimer akan terus ditambahkan ke dalam kopi.

Tabel 5.13 Pengujian Kopi untuk Jenis Kopi *very light*

No	Jenis Kopi	Output Crisp	Kebenaran Jenis Kopi	Keterangan
1	<i>Very light</i>	0.830715	BENAR	-
2	<i>Very light</i>	0.830998	BENAR	-
3	<i>Very light</i>	0.830768	BENAR	-
4	<i>Very light</i>	-	SALAH	Motor terus berputar sampai tak terbatas sehingga <i>output</i> di luar <i>range output crisp very light</i>
5	<i>Very light</i>	0.831643	BENAR	-
6	<i>Very light</i>	0.831296	BENAR	-
7	<i>Very light</i>	0.831468	BENAR	-
8	<i>Very light</i>	0.831658	BENAR	-
9	<i>Very light</i>	0.713889	BENAR	-
10	<i>Very light</i>	0.703741	BENAR	-
11	<i>Very light</i>	0.710266	BENAR	-

12	<i>Very light</i>	0.720768	BENAR	-
13	<i>Very light</i>	0.831253	BENAR	-
14	<i>Very light</i>	0.831591	BENAR	-
15	<i>Very light</i>	0.72453	BENAR	-
16	<i>Very light</i>	0.701372	BENAR	-
17	<i>Very light</i>	0.707857	BENAR	-
18	<i>Very light</i>	0.710034	BENAR	-
19	<i>Very light</i>	0.730764	BENAR	-
20	<i>Very light</i>	-	SALAH	Motor terus berputar sampai tak terbatas sehingga <i>output</i> di luar <i>range output crisp very light</i>
Persentase Kebenaran Jenis Kopi			18/20 X 100% = 90%	
Persentase Kesalahan Jenis Kopi			2/20 X 100% = 10%	

Pada tabel 5.13 di atas ditunjukkan bahwa untuk jenis kopi *very light*, masih terdapat 2 kali kesalahan pengenalan warna dari 20 kali pengujian. Kesalahan ini dapat terjadi karena nilai *input*FIS berada di luar rentang yang diujikan, sehingga *output crisp* yang dihasilkan tidak lagi berada pada rentang *outputvery light*. Kesalahan ini menyebabkan pengaduk kopi akan terus mengaduk kopi, dan penutup saluran krimer akan terus membuka dan menutup sehingga krimer akan terus ditambahkan ke dalam kopi.

Tabel 5.14 Performansi total mesin pengaduk kopi otomatis dengan FIS Sugeno.

No	Jenis Kopi	Persentase Kebenaran Pengujian	Persentase Kesalahan Pengujian
1	<i>Very dark</i>	90%	10%
2	<i>Dark</i>	100%	0%
3	<i>Middle</i>	100%	0%
4	<i>Light</i>	85%	15%
5	<i>Very light</i>	90%	10%
Persentase Rata-rata		$465/500 \times 100\% = 93\%$	$35/500 \times 100\% = 7\%$

Dari tabel 5.14 di atas ditunjukkan persentase rata-rata keberhasilan adalah sebesar 93% yang secara keseluruhan sudah baik, sedangkan persentase rata-rata kesalahan adalah sebesar 7%. Kesalahan terbanyak terjadi pada jenis kopi Ringan/ *light* yaitu sebanyak 3 kali dari 20 kali pengujian. Total kesalahan terjadi sebanyak 7 kali dari 100 pengujian. Dari kesalahan-kesalahan yang terjadi dapat disimpulkan bahwa hal ini dapat terjadi karena nilai *input* FIS yang terkadang melebihi rentang yang telah disediakan, sehingga *output* yang dihasilkan melebihi *range* nilai *output* FIS.

5.2.3.3 Analisa hasil pengujian FIS Mamdani dan FIS Sugeno

Pada penelitian ini digunakan FIS Mamdani dan FIS Sugeno sebagai sistem pengambil keputusan. Pada pengujian sistem dengan FIS Mamdani didapatkan persentase performansi total sebesar 92%, sedangkan pengujian sistem dengan menggunakan FIS Sugeno didapatkan persentase performansi total sebesar 93%. Pada pengujian dengan FIS Mamdani didapatkan kesalahan pengenalan warna kopi sebanyak 8 kali dari 100 kali pengujian sistem dengan

kesalahan terbanyak terjadi pada pengujian jenis kopi *light*, yaitu sebanyak 3 kali. Sedangkan pada FIS Sugeno didapatkan kesalahan pengenalan warna kopi sebanyak 7 kali dari 100 kali pengujian sistem dengan kesalahan terbanyak terjadi pada pengujian kopi *light*, yaitu sebanyak 3 kali.

Pada pengujian FIS Mamdani, hasil pengujian sistem dengan performansi 100% hanya terdapat pada pengujian jenis kopi *very dark*. Sedangkan pada pengujian FIS Sugeno, hasil pengujian sistem dengan performansi 100% terdapat pada pengujian jenis kopi *dark* dan *middle*.

Berdasarkan hasil pengujian ini, walaupun secara keseluruhan hanya terdapat perbedaan kesalahan pengujian sebanyak satu kali, dapat dilihat bahwa sistem dengan FIS Sugeno memiliki performansi yang lebih baik dibandingkan sistem dengan FIS Mamdani.

Hal ini sesuai dengan teori yang mengatakan bahwa FIS Sugeno lebih unggul dibandingkan FIS Mamdani^[6]. Pada FIS Sugeno, proses fuzzifikasinya menggunakan metode *weighted average*, sedangkan pada FIS Mamdani menggunakan metode *centroid*. Pada metode *centroid*, fuzzifikasi dilakukan dengan menggunakan rumus integral dan pada metode *weighted average*, fuzzifikasi dilakukan dengan menggunakan rumus rata-rata. Perhitungan dengan menggunakan rumus integral lebih kompleks dibandingkan dengan rumus rata-rata dan membutuhkan penyederhanaan-penyederhanaan angka dalam perhitungannya. Hal ini dapat mengurangi keakuratan hasil perhitungan. Hal inilah yang dapat menyebabkan performansi FIS Mamdani lebih buruk dibandingkan dengan FIS Sugeno.

BAB VI

PENUTUP

6.1 Kesimpulan

Berdasarkan analisa terhadap hasil yang didapatkan maka dapat disimpulkan sebagai berikut:

1. Sistem yang dibuat berdasarkan pengenalan warna dan prinsip penarikan kesimpulan dengan logika *fuzzy inference system* (FIS) model Mamdani dan Sugeno dalam penelitian ini telah dapat melakukan otomatisasi pada pergerakan pengaduk kopi dan katup krimer berdasarkan warna kopi, dimana *output* yang dihasilkan oleh system ini telah sama dengan *input* nya.
2. Pada penelitian ini, sistem pengambilan keputusan dengan menggunakan FIS Mamdani menghasilkan performansi sebesar 92% sedangkan sistem pengambilan keputusan dengan menggunakan FIS Sugeno menghasilkan performansi sebesar 93%. Sehingga dapat disimpulkan bahwa pada sistem ini FIS Sugeno memiliki performa yang lebih baik dibandingkan FIS Mamdani.
3. Sistem sensor kamera berdasarkan warna yang dilakukan oleh alat pengaduk kopi otomatis terhadap objek berupa larutan kopi dapat bekerja sesuai dengan jenis kopi yang dipilih.

6.2 Saran

Untuk tujuan penelitian dan pengembangan sistem ini selanjutnya, penulis menyarankan pada mesin pengaduk kopi ini ditambahkan saluran bubuk kopi langsung sehingga kondisi awal kopi tidak selalu harus

dimulai dari kondisi *very dark* dan mesin lebih fleksibel terhadap ukuran gelas yang dipakai.



DAFTAR KEPUSTAKAAN

- [1] Aditya. 2009. *Sistem Kamera Penjejak Objek Berdasarkan Warna Menggunakan Motor Servo DC*. Universitas Andalas. Padang.
- [2] Ayu, Prima Shanti Ida. *Lecnote Manusia, Komputer dan Interaksi*.
Sumber:<http://santiw.staff.gunadarma.ac.id/Downloads/files/7676/Lecnote+Manusia,+Komputer+dan+Interaksi.doc>. Diakses pada Senin, 27/12/2010. 11.05 am.
- [3] Francois, Alexandre R.J. and Gerard D. Medioni. 1999. "*Adaptive Color Background Modeling for Real-Time Segmentation of Video Streams*". Los Angeles: University of Southern California.
- [4] <http://elib.unikom.ac.id/download.php?id=42933>. Diakses pada Kamis, 03/03/2011. 13.47 pm.
- [5] <http://www.uswahtech.uswah.net/berita-217-logika-fuzzy.html>. Diakses pada Kamis, 03/03/2011. 13.48 pm.
- [6] Effendy, Nazrul. 2008. *Aplikasi Kontrol Logika Fuzzy pada sistem Tracking Matahari Panel Photovoltaic*. Universitas Gadjah Mada. Yogyakarta.
- [7] <http://www.ilmu.8k.com/pengetahuan/stepper.htm> . Diakses pada Rabu, 16/05/2012. 10.27 am.
- [8] <http://www.ilmu.8k.com/pengetahuan/stepper.htm> . Diakses pada Kamis, 25/8/2011. 9.47 pm.

- [9] <http://www.staff.ui.ac.id/internal/040603019/material/DCMotorPaperandQA.pdf>. Diakses pada Selasa, 17/04/2012. 10.23 pm.
- [10] <http://www.duniaelektronika.blogspot.com/2007/09/mikrokontroler-atmega8535.html> . Diakses pada Rabu, 18/04/2012. 1.54 am.
- [11] *IT Telkom Library. Aktuator.* 2010. Sumber :
http://www.ittelkom.ac.id/library/index.php?option=com_content&view=article&id=690:aktuator&catid=16:mikroprocessorkontroller&Itemid=15.
Diakses pada Kamis, 19/01/2011. 12.39 pm.
- [12] Kurnia, Rahmadi, Silvaningrum Nurhadi. *Deteksi Obyek Berbasis Warna Dan Ukuran Dengan Bantuan Interaksi Komputer-Mamusia.* Sumber :
http://repository.akprind.ac.id/sites/files/conference-paper/2008/kurnia_2084.pdf . Diakses pada Senin, 22/8/2011. 2.25 pm.
- [13] Kurnia, Rahmadi.2009. "*Penjejak Target Benda Pada Gerakan Linier Berdasarkan Warna*". Universitas Andalas. Padang.
- [14] Kurniawan, Obeth. *Segmentasi Citra Sel Kanker dengan Fuzzy Inference System Menggunakan Matlab.* Sumber:
http://docs.google.com/viewer?a=v&q=cache:OzCuXSYNr48J:www.ta.trisakti.ac.id/ta/sites/default/files/proposal%2520TA_3.doc+proses+pengenalan+citra+dengan+FIS&hl=id&gl=id&pid=bl&srcid=ADGEESgnIxXQWtcgy6iQ-bFF7CZ3Vd04LQwyoXGT6E3Cm94AmZNpA-xiPtIc-VjHBxUE5L29Zm0DoOzG_BLBVJQOOAEqGnNRpQZj2hK88iDW50DRnEJOOfRodzQc-

[8OpFI5jw4fOM5PGy&sig=AHIEtbRHOPllkGpulpm3P5VVVQ1cxrOdKg](http://www.digi-ware.com/file/27-EE-08.pdf)

. Diakses pada Kamis, 25/8/2011.6.56 am.

[15] Lestari, Dwi Rilli. *Pengenalan Penyakit Darah dengan Logika Fuzzy*.

Sumber : <http://www.digi-ware.com/file/27-EE-08.pdf>. Diakses pada Minggu, 14/08/2011. 4.23 am.

[16] Leylana, Rosa. *Pengaruh Kopi Terhadap Asam Urat Darah*. Sumber :

[http://docs.google.com/viewer?a=v&q=cache:ntsqa_poUJ:eprints.undip.ac](http://docs.google.com/viewer?a=v&q=cache:ntsqa_poUJ:eprints.undip.ac.id/19270/1/Rosa_Lelyana.pdf+kandungan+kopi&hl=id&gl=id&pid=bl&srcid=ADGEESg-)
[.id/19270/1/Rosa_Lelyana.pdf+kandungan+kopi&hl=id&gl=id&pid=bl&src](http://docs.google.com/viewer?a=v&q=cache:ntsqa_poUJ:eprints.undip.ac.id/19270/1/Rosa_Lelyana.pdf+kandungan+kopi&hl=id&gl=id&pid=bl&srcid=ADGEESg-)
[id=ADGEESg-](http://docs.google.com/viewer?a=v&q=cache:ntsqa_poUJ:eprints.undip.ac.id/19270/1/Rosa_Lelyana.pdf+kandungan+kopi&hl=id&gl=id&pid=bl&srcid=ADGEESg-)

[ENSzPeSeUAu2QaDfdbm5aHcHr3QawKJK2pUydGh8UupeN5ATRd_ZE8](http://docs.google.com/viewer?a=v&q=cache:ntsqa_poUJ:eprints.undip.ac.id/19270/1/Rosa_Lelyana.pdf+kandungan+kopi&hl=id&gl=id&pid=bl&srcid=ADGEESg-)
[Sq6Kgoll-](http://docs.google.com/viewer?a=v&q=cache:ntsqa_poUJ:eprints.undip.ac.id/19270/1/Rosa_Lelyana.pdf+kandungan+kopi&hl=id&gl=id&pid=bl&srcid=ADGEESg-)

[UQW3hH8zYhUgcHOxm80So1s0T8EafFpF47yOJO873EHt9IFNZZWct](http://docs.google.com/viewer?a=v&q=cache:ntsqa_poUJ:eprints.undip.ac.id/19270/1/Rosa_Lelyana.pdf+kandungan+kopi&hl=id&gl=id&pid=bl&srcid=ADGEESg-)

[FJuA5WDyeJyyT0n&sig=AHIEtbTE0AH6xqD_ang8jc0SyzGjIaPe_A&pli](http://docs.google.com/viewer?a=v&q=cache:ntsqa_poUJ:eprints.undip.ac.id/19270/1/Rosa_Lelyana.pdf+kandungan+kopi&hl=id&gl=id&pid=bl&srcid=ADGEESg-)

[=1](http://docs.google.com/viewer?a=v&q=cache:ntsqa_poUJ:eprints.undip.ac.id/19270/1/Rosa_Lelyana.pdf+kandungan+kopi&hl=id&gl=id&pid=bl&srcid=ADGEESg-) . Diakses pada : Kamis, 25/8/2011. 6.13 am.

[17] Naba, Agus. 2009. "*Belajar. Cepat Fuzzy Logic menggunakan MATLAB*".

Yogyakarta: Penerbit ANDI.

[18] Purnomo, Dwi. *Sistem Pakar Fuzzy Penentuan dan Peningkatan Kualitas*

Manggis

Sumber:

[http://pustaka.unpad.ac.id/wp-](http://pustaka.unpad.ac.id/wp-content/uploads/2009/09/sistem_pakar_fuzzy_penentuan_dan_penigkatan_kualitas_manggis.pdf)
[content/uploads/2009/09/sistem_pakar_fuzzy_penentuan_dan_penigkatan_k](http://pustaka.unpad.ac.id/wp-content/uploads/2009/09/sistem_pakar_fuzzy_penentuan_dan_penigkatan_kualitas_manggis.pdf)
[ualitas_manggis.pdf](http://pustaka.unpad.ac.id/wp-content/uploads/2009/09/sistem_pakar_fuzzy_penentuan_dan_penigkatan_kualitas_manggis.pdf) . Diakses pada Kamis, 25/8/2011. 7.31 pm.

[19] Yao, Wang, J.Ostermann & Y.Q.Zhang. 2002. *Video Processing and*

Communications. New Jersey: Prentice Hall.

Lampiran A, Program pada GUI Pengaduk Kopi Otomatis

```
function varargout = PENGADUKKOIFIX(varargin)
% PENGADUKKOIFIX M-file for PENGADUKKOIFIX.fig
%   PENGADUKKOIFIX, by itself, creates a new PENGADUKKOIFIX or
%   raises the existing
%   singleton*.
%
%   H = PENGADUKKOIFIX returns the handle to a new
%   PENGADUKKOIFIX or the handle to
%   the existing singleton*.
%
%   PENGADUKKOIFIX('CALLBACK', hObject,eventData,handles,...)
%   calls the local
%   function named CALLBACK in PENGADUKKOIFIX.M with the given
%   input arguments.
%
%   PENGADUKKOIFIX('Property','Value',...) creates a new
%   PENGADUKKOIFIX or raises the
%   existing singleton*. Starting from the left, property value
%   pairs are
%   applied to the GUI before PENGADUKKOIFIX_OpeningFcn
%   gets called. An
%   unrecognized property name or invalid value makes property
%   application
%   stop. All inputs are passed to PENGADUKKOIFIX_OpeningFcn
%   via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
%   only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES.

% Edit the above text to modify the response to help PENGADUKKOIFIX

% Last Modified by GUIDE v2.5 08-Apr-2012 13:51:14

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',   @PENGADUKKOIFIX_OpeningFcn,
                  ...
                  'gui_OutputFcn',    @PENGADUKKOIFIX_OutputFcn, ...
                  'gui_LayoutFcn',    [], ...
                  'gui_Callback',     []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
```

```

else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
%*****
%           PROGRAM FORM GUI PENGADUK KOPI OTOMATIS
%           OLEH:
%           ILFIYANTRI INTYAS
%           07 175 049
%           PEMBIMBING : Dr Eng. Rahmadi Kurnia
%*****

% --- Executes just before PENGADUKKOPIFIX is made visible.
function PENGADUKKOPIFIX_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to PENGADUKKOPIFIX (see
VARARGIN)

% Choose default command line output for PENGADUKKOPIFIX
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes PENGADUKKOPIFIX wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = PENGADUKKOPIFIX_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in DARK.
function DARK_Callback(hObject, eventdata, handles)
% hObject    handle to DARK (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

myform = guidata (gcbo);
daqhwinfo('parallel');
dio = digitalio('parallel','LPT1');
addline(dio,0:7,0,'out');
putvalue(dio,253);
vid=videoinput('winvideo',1,'RGB24_160x120');
set(vid, 'FramesPerTrigger', Inf);
vid.FrameGrabInterval = 29;
preview(vid);
pause(15);
data=getsnapshot(vid);
rgb=imcrop(data,[1 56 1 50]);
red=rgb(:,:,1);
green=rgb(:,:,2);
blue=rgb(:,:,3);
r=mean2(red(:));
g=mean2(green(:));
b=mean2(blue(:));
num=0.5*((r-g)+(r-b));
den=sqrt((r-g).^2+(r-b).*(g-b));
theta=acos(num./(den+eps));
H=theta;
H(b>g)=2*pi-H(b>g);
H=H/(2*pi);
num=min(min(r,g),b);
den=r+g+b;
den(den==0)=eps;
S=1-3.*num./den;
H(S==0)=0;
I=(r+g+b)/3;
input=[H]
a=readfis('coffee.fis');
output=evalfis(input,a)
hasil=KonversiDataKopi(output);
while(vid.FramesAcquired<=100);
    if hasil~=1
        capture
    elseif hasil==1
        dio = digitalio('parallel','LPT1');
        addline(dio,0:7,0,'out');
        putvalue(dio,251);
        set(myform.text9,'string',input);
        set(myform.text11,'string',output);
        set(myform.text15,'string',I);
        imshow(data);
        break
    end
end
end

```

```

% --- Executes on button press in LIGHT.
function LIGHT_Callback(hObject, eventdata, handles)

```

```

% hObject    handle to LIGHT (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
myform = guidata(gcbo);
daqhwinfo('parallel');
dio = digitalio('parallel','LPT1');
addline(dio,0:7,0,'out');
putvalue(dio,253);
vid=videoinput('winvideo',1,'RGB24_160x120');
set(vid, 'FramesPerTrigger', Inf);
vid.FrameGrabInterval = 29;
preview(vid);
pause(15);
data=getsnapshot(vid);
rgb=imcrop(data,[1 56 1 50]);
red=rgb(:,:,1);
green=rgb(:,:,2);
blue=rgb(:,:,3);
r=mean2(red(:))
g=mean2(green(:))
b=mean2(blue(:))
num=0.5*((r-g)+(r-b));
den=sqrt((r-g).^2+(r-b).*(g-b));
theta=acos(num./(den+eps));
H=theta;
H(b>g)=2*pi-H(b>g);
H=H/(2*pi);
num=min(min(r,g),b);
den=r+g+b;
den(den==0)=eps;
S=1-3.*num./den;
H(S==0)=0;
I=(r+g+b)/3;
input=[H]
a=readfis('coffee.fis');
output=evalfis(input,a)
hasil=KonversiDataKopi(output);
while(vid.FramesAcquired<=100);
    if hasil~=3
        capture
    elseif hasil==3
        dio = digitalio('parallel','LPT1');
        addline(dio,0:7,0,'out');
        putvalue(dio,251);
        set(myform.text9,'string',input);
        set(myform.text11,'string',output);
        set(myform.text15,'string',I);
        imshow(data);
        break
    end
end
end

```

```

% --- Executes on button press in EXIT.
function EXIT_Callback(hObject, eventdata, handles)
% hObject    handle to EXIT (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close(handles.figure1)

% --- Executes on button press in VERYDARK.
function VERYDARK_Callback(hObject, eventdata, handles)
% hObject    handle to VERYDARK (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
myform = guidata(gcbo)
daqhwinfo('parallel');
dio = digitalio('parallel','LPT1');
addline(dio,0:7,0,'out');
putvalue(dio,254);
vid=videoinput('winvideo',1,'RGB24_160x120');
set(vid, 'FramesPerTrigger', Inf);
vid.FrameGrabInterval = 29;
preview(vid);
pause(15);
data=getsnapshot(vid);
rgb=imcrop(data,[1 56 1 50]);
red=rgb(:,:,1);
green=rgb(:,:,2);
blue=rgb(:,:,3);
r=mean2(red(:))
g=mean2(green(:))
b=mean2(blue(:))
num=0.5*((r-g)+(r-b));
den=sqrt((r-g).^2+(r-b).*(g-b));
theta=acos(num./(den+eps));
H=theta;
H(b>g)=2*pi-H(b>g);
H=H/(2*pi);
num=min(min(r,g),b);
den=r+g+b;
den(den==0)=eps;
S=1-3.*num./den;
H(S==0)=0;
I=(r+g+b)/3;
input=[H]
a=readfis('coffee.fis');
output=evalfis(input,a)
hasil=KonversiDataKopi(output);
while(vid.FramesAcquired<=100);
    if hasil~=0
        capture
    elseif hasil==0
        dio = digitalio('parallel','LPT1');
        addline(dio,0:7,0,'out');
        putvalue(dio,251);
    end
end

```

```

        imshow(data);
        set(myform.text9, 'string', input);
        set(myform.text11, 'string', output);
        set(myform.text15, 'string', I);
        imshow(data);
        break
    end
end
end

```

```

% --- Executes on button press in MIDDLE.
function MIDDLE_Callback(hObject, eventdata, handles)
% hObject      handle to MIDDLE (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
myform = guidata(gcbo)
daqhwinfo('parallel');
dio = digitalio('parallel', 'LPT1');
addline(dio, 0:7, 0, 'out');
putvalue(dio, 253);
vid=videoinput('winvideo', 1, 'RGB24_160x120');
set(vid, 'FramesPerTrigger', Inf);
vid.FrameGrabInterval = 29;
preview(vid);
pause(15);
data=getsnapshot(vid);
rgb=imcrop(data, [1 56 1 50]);
red=rgb(:, :, 1);
green=rgb(:, :, 2);
blue=rgb(:, :, 3);
r=mean2(red(:))
g=mean2(green(:))
b=mean2(blue(:))
num=0.5*((r-g)+(r-b));
den=sqrt((r-g).^2+(r-b).*(g-b));
theta=acos(num./(den+eps));
H=theta;
H(b>g)=2*pi-H(b>g);
H=H/(2*pi);
num=min(min(r,g),b);
den=r+g+b;
den(den==0)=eps;
S=1-3.*num./den;
H(S==0)=0;
I=(r+g+b)/3;
input=[H]
a=readfis('coffee.fis');
output=evalfis(input,a)
hasil=KonversiDataKopi(output);
while(vid.FramesAcquired<=100);
    if hasil~=2
        capture
    elseif hasil==2
        dio = digitalio('parallel', 'LPT1');
    end
end

```

```

        addline(dio,0:7,0,'out');
        putvalue(dio,251);
        set(myform.text9,'string',input);
        set(myform.text11,'string',output);
        set(myform.text15,'string',I);
        imshow(data);
        break
    end
end

```

```

% --- Executes on button press in VERYLIGHT.
function VERYLIGHT_Callback(hObject, eventdata, handles)
% hObject      handle to VERYLIGHT (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
myform = guidata(gcbo)
daqhwinfo('parallel');
dio = digitalio('parallel','LPT1');
addline(dio,0:7,0,'out');
putvalue(dio,252);
vid=videoinput('winvideo',1,'RGB24_160x120');
set(vid,'FramesPerTrigger', Inf);
vid.FrameGrabInterval = 29;
preview(vid);
pause(15);
data=getsnapshot(vid);
rgb=imcrop(data,[1 56 1 50]);
red=rgb(:,:,1);
green=rgb(:,:,2);
blue=rgb(:,:,3);
r=mean2(red(:))
g=mean2(green(:))
b=mean2(blue(:))
num=0.5*((r-g)+(r-b));
den=sqrt((r-g).^2+(r-b).*(g-b));
theta=acos(num./(den+eps));
H=theta;
H(b>g)=2*pi-H(b>g);
H=H/(2*pi);
num=min(min(r,g),b);
den=r+g+b;
den(den==0)=eps;
S=1-3.*num./den;
H(S==0)=0;
I=(r+g+b)/3;
input=[H I]
a=readfis('coffee.fis');
output=evalfis(input,a)
hasil=KonversiDataKopi(output);
while(vid.FramesAcquired<=100);
    if hasil~=4
        capture
    end
end

```

```

elseif hasil==4
    dio = digitalio('parallel','LPT1');
    addline(dio,0:7,0,'out');
    putvalue(dio,251);
    set(myform.text9,'string',input);
    set(myform.text11,'string',output);
    set(myform.text15,'string',I);
    imshow(data);
    break
end
end

```

```

function INPUTFIS_Callback(hObject, eventdata, handles)
% hObject      handle to TEKSINPUT (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of TEKSINPUT as text
%         str2double(get(hObject,'String')) returns contents of
TEKSINPUT as a double

```

```

% --- Executes during object creation, after setting all properties.
function INPUTFIS_CreateFcn(hObject, eventdata, handles)
% hObject      handle to TEKSINPUT (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

```

```

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

% --- Executes on button press in text9.
function text9_Callback(hObject, eventdata, handles)
% hObject      handle to text3 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

```

```

% --- Executes on button press in text11.
function text11_Callback(hObject, eventdata, handles)
% hObject      handle to text3 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

```

```

% --- Executes on button press in text15.
function text15_Callback(hObject, eventdata, handles)
% hObject    handle to text3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

function OUTPUTFIS_Callback(hObject, eventdata, handles)
% hObject    handle to TEKSOUTPUT (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of TEKSOUTPUT as
text
%         str2double(get(hObject,'String')) returns contents of
TEKSOUTPUT as a double

% --- Executes during object creation, after setting all properties.
function OUTPUTFIS_CreateFcn(hObject, eventdata, handles)
% hObject    handle to TEKSOUTPUT (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

LAMPIRAN B, Program coffee.FIS

```
[System]
Name='coffee'
Type='mamdani'
Version=2.0
NumInputs=1
NumOutputs=1
NumRules=5
AndMethod='min'
OrMethod='max'
ImpMethod='min'
AggMethod='max'
DefuzzMethod='centroid'

[Input1]
Name='H'
Range=[0 0.16]
NumMFs=5
MF1='verydark': 'trimf', [0.117 0.122962962962963 0.16]
MF2='middle': 'trimf', [0.0794 0.084021164021164 0.104]
MF3='verylight': 'trimf', [0.0005 0.0273015873015873 0.0717]
MF4='light': 'trimf', [0.0599 0.064973544973545 0.087]
MF5='dark': 'trimf', [0.0933 0.0967 0.127619047619048]

[Output1]
Name='jenis_kopi'
Range=[0 1]
NumMFs=5
MF1='very_dark': 'trimf', [-0.391354497354497 0.00823550264550256
0.199645502645503]
MF2='middle': 'trimf', [0.236862433862434 0.433862433862434
0.638888888888889]
MF3='very_light': 'trimf', [0.674084656084656 0.819084656084656
0.993084656084656]
MF4='dark': 'trimf', [0.00401 0.218253968253968 0.448]
MF5='light': 'trimf', [0.469989417989418 0.641534391534391
0.810989417989418]

[Rules]
1, 1 (1) : 1
2, 2 (1) : 1
3, 3 (1) : 1
4, 5 (1) : 1
5, 4 (1) : 1
```

Lampiran C, Program coffeesugeno.fis

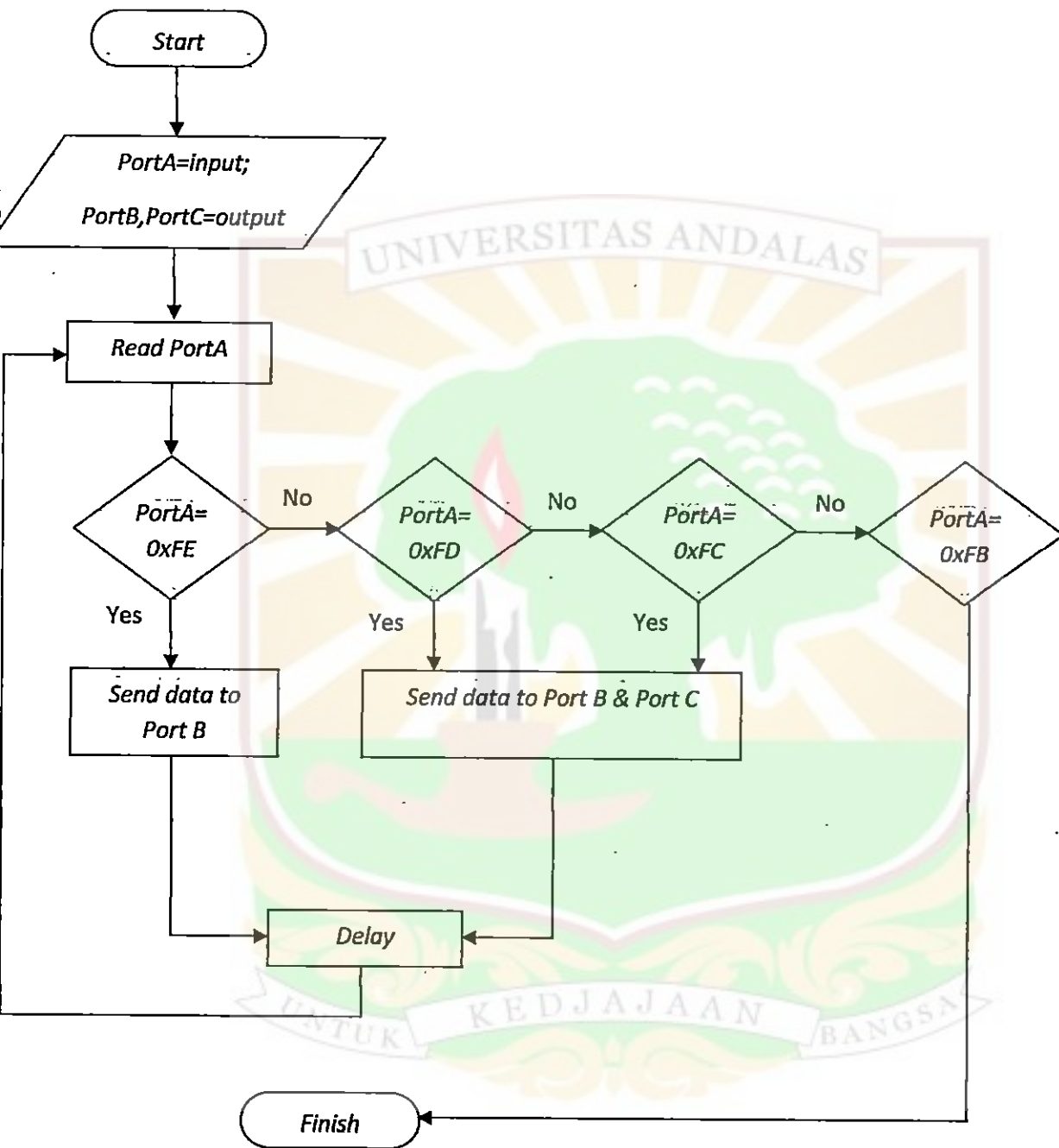
```
[System]
Name='coffeesugeno'
Type='sugeno'
Version=2.0
NumInputs=1
NumOutputs=1
NumRules=5
AndMethod='prod'
OrMethod='probor'
ImpMethod='prod'
AggMethod='sum'
DefuzzMethod='wtaver'
```

```
[Input1]
Name='H'
Range=[0 0.16]
NumMFs=5
MF1='verylight': 'trimf', [0.0005 0.0273 0.0717]
MF2='light': 'trimf', [0.0599 0.06497 0.087]
MF3='middle': 'trimf', [0.0794 0.08402 0.104]
MF4='dark': 'trimf', [0.0933 0.0967 0.1276]
MF5='verydark': 'trimf', [0.117 0.123 0.16]
```

```
[Output1]
Name='jeniskopi'
Range=[0 1]
NumMFs=5
MF1='verydark': 'constant', [0.1]
MF2='dark': 'constant', [0.09]
MF3='middle': 'constant', [0.08]
MF4='light': 'constant', [0.06]
MF5='verylight': 'constant', [0.02]
```

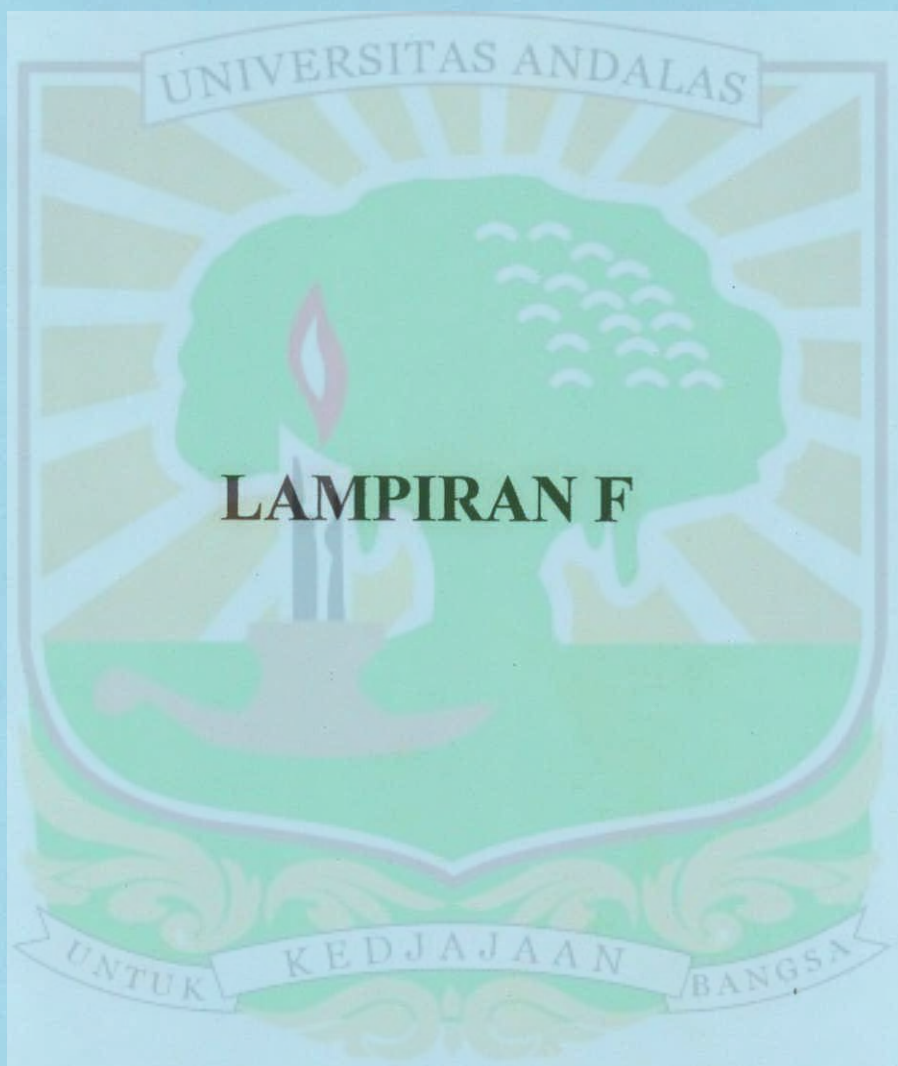
```
[Rules]
1, 5 (1) : 1
2, 4 (1) : 1
3, 3 (1) : 1
4, 2 (1) : 1
5, 1 (1) : 1
```

Lampiran D, Flowchart Mikrokontroler



Lampiran E , Gambar Alat Pengaduk Kopi





LAMPIRAN F

Lampiran F, Gambar Rangkaian Sistem.

