

# Sistem Pengendali Pemanas Pemanggang Kopi Menggunakan Logika *Fuzzy*

## *Heater Coffee Roaster Controller System Using Fuzzy Logic*

Eko Joni Pristianto, Hana Arisesa, dan Arief Nur Rahman  
Pusat Penelitian Elektronika dan Telekomunikasi – LIPI  
Komplek LIPI Gd 10 Lt 2, JL. Sangkuriang, Bandung 40135  
Email: [ekojonip5@gmail.com](mailto:ekojonip5@gmail.com)

---

### **Abstract**

*Coffee roaster is commonly used for transforming the chemical and physical properties of green coffee beans into roasted coffee products. The working principle of coffee roaster is roasting raw coffee beans in a rotating drum using high temperature for a limited period of time. One of roaster heater element is using electric system which is operated manually using switch or semi-automatic timer by an operator. Good taste coffee is determined by excellent roasting process by an expert operator. Therefore, in this article a heater element control concept is introduced using fuzzy logic system. Fuzzy rules are build based on expertise or experience of expert operator. Logic fuzzy is implemented with microcontroller using C. The purpose of this work is to develop fuzzy controller system for automatic heating of coffee roasting process. The output of this research is a prototype which is represent of heater coffee roaster controller using fuzzy logic.*

**Keywords:** coffee roaster, fuzzy, microcontroller

### **Abstrak**

Mesin pemanggang kopi merupakan sebuah mesin yang digunakan untuk menyangrai biji kopi agar matang, sehingga siap untuk diproses lebih lanjut. Prinsip kerja mesin ini adalah produk dipanaskan dalam ruang sangrai yang berputar dengan suhu tertentu, sehingga pemanasan dapat merata. Salah satu jenis pemanas pemanggang kopi adalah elemen pemanas listrik, dimana sistem kerjanya masih banyak dikendalikan secara manual dengan saklar atau semi otomatis menggunakan *timer* yang dioperasikan oleh seorang operator. Proses pemanggangan (*roasting*) kopi sangat menentukan cita rasa kopi, sehingga dibutuhkan seorang operator yang ahli di bidang ini. Pada paper ini akan didesain sebuah sistem pengendali pemanas pemanggang kopi dengan logika fuzzy. Aturan-aturan fuzzy yang akan dibangun, didasarkan atas keahlian atau pengalaman seorang operator. Logika *fuzzy* yang dihasilkan akan diaplikasikan pada sebuah mikrokontroler dengan menggunakan pemrograman bahasa C. Tujuan dari penelitian ini adalah membangun sebuah sistem pengendali pada pemanas pemanggang dengan logika *fuzzy*, sehingga proses *roasting* biji kopi dapat bekerja secara otomatis. Hasil penelitian ini adalah sebuah *prototype* yang merepresentasikan sistem kerja pengendali pemanas pemanggang kopi menggunakan logika *fuzzy*.

**Kata kunci:** pemanggang kopi, *fuzzy*, mikrokontroler

---

### **1. Pendahuluan**

Pemanggang kopi adalah sebuah mesin yang digunakan untuk menyangrai biji kopi agar matang dan kering sehingga siap untuk diproses lebih lanjut. Prinsip kerja mesin pemanggang ini adalah produk dipanaskan dalam ruang sangrai yang berputar dengan suhu tertentu, sehingga pemanasan dapat merata. Salah satu jenis pemanas mesin pemanggang adalah elemen pemanas listrik,

dimana sistem kerjanya masih dikendalikan secara manual dengan saklar atau semi-otomatis menggunakan *timer* yang dioperasikan oleh seorang operator [1].

Suhu pemanasan pada pemanggang kopi berkisar antara 0-200°C. Terdapat 16 tahapan warna biji kopi, dari biji kopi mentah sampai matang, yaitu mulai dari warna hijau, kuning, agak coklat, coklat, dan hitam. Gambar 1 menunjukkan standar kematangan biji kopi [2].



Gambar 1. Standar kematangan biji kopi [2].

Paper ini membahas desain sebuah sistem pengendali elemen pemanas pemanggang kopi dengan logika *fuzzy*. Aturan-aturan *fuzzy* yang akan dibangun, didasarkan atas keahlian atau pengalaman seorang operator. Keuntungan menggunakan metode ini akan didapatkan kualitas biji kopi hasil sangrai lebih seragam, serta tidak tergantung dari ada tidaknya operator ahli. Pemilihan menggunakan metode *fuzzy* karena algoritma ini lebih mendekati pola pikir manusia, proses perhitungan sederhana dan respon yang cepat untuk diterapkan pada sistem kendali.

Logika *fuzzy* yang dihasilkan akan diaplikasikan pada sebuah modul mikrokontroler *Arduino ATmega 328* dengan menggunakan pemrograman bahasa C. *Toolbox Fuzzy Inference System (FIS)* editor pada Matlab digunakan untuk memverifikasi perhitungan logika *fuzzy* pada mikrokontroler.

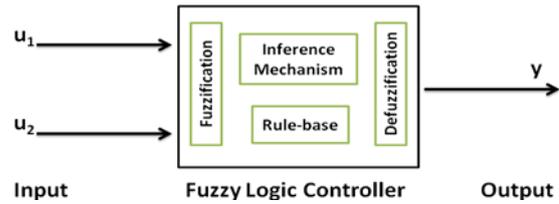
Batasan masalah pada perancangan ini adalah rancangan hanya dilakukan untuk mengendalikan elemen pemanas pada mesin pemanggang kopi berdasarkan tingkat warna kopi dan temperatur ruang sangrai. Proses logika *fuzzy* yang digunakan adalah model Mamdani dengan defuzzifikasi *Mean of Maximum (MOM)*. *Prototype* yang dibuat, hanya akan mengilustrasikan prinsip kerja pengendali elemen pemanas dari mesin pemanggang kopi.

Tujuan dari penelitian ini adalah mencari alternatif solusi dalam pengendalian pemanas pemanggang kopi, yaitu dengan cara mengendalikan tingkat pemanasan elemen pemanas listrik pada mesin pemanggang kopi menggunakan logika *fuzzy*, sehingga proses pemanggangan biji kopi dapat terkendali secara otomatis dengan aturan-aturan *fuzzy* yang didasarkan pada keahlian seorang operator.

## 2. Dasar Teori

### 2.1. Logika Fuzzy

Logika *fuzzy* diciptakan untuk mengurangi kekakuan dari logika kendali biner yang berlogika 1 dan 0. Pada logika *fuzzy* berlaku logika antara 1 dan 0, logika *fuzzy* pada umumnya terdiri dari *fuzzification*, *membership function*, *rule* dan *defuzzification* [3]. Gambar 2 menunjukkan diagram blok dari *fuzzy Logic Controller* [3].

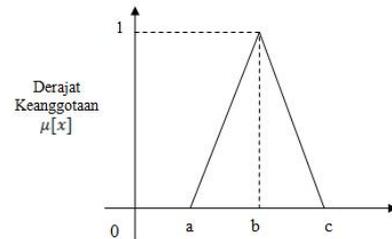


Gambar 2. Blok diagram *Fuzzy Logic Controller*[3].

Fungsi keanggotaan logika *fuzzy* digunakan untuk menghitung derajat keanggotaan suatu himpunan *fuzzy*. Fungsi keanggotaan yang biasa digunakan dalam penalaran logika *fuzzy*, diantaranya:

#### 1. Kurva Segitiga

Kurva segitiga pada dasarnya merupakan gabungan antara 2 garis (*linear*). Nilai-nilai disekitar *b* memiliki derajat keanggotaan turun cukup tajam (menjauhi 1). Sebagaimana Gambar 3.

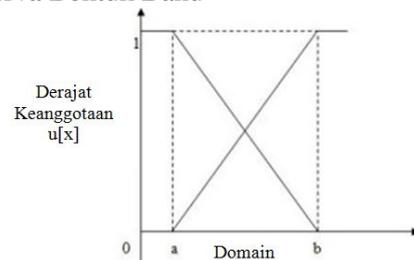


Gambar 3. Kurva segitiga

Representasi fungsi keanggotaan untuk kurva segitiga adalah sebagai berikut [3]:

$$\text{segitiga}(x, a, b, c) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{c-x}{c-b}, & b \leq x \leq c \\ 0, & c \leq x \end{cases} \quad (1)$$

#### 2. Kurva Bentuk Bahu



Gambar 4. Kurva Bentuk Bahu

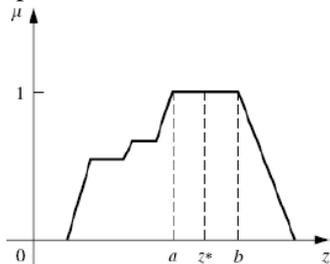
Representasi fungsi keanggotaan untuk kurva bahu adalah sebagai berikut [3]:

$$\mu[x, a, b] = \begin{cases} 0; & x \leq a \\ \frac{b-x}{b-a}; & a \leq x \leq b \\ 1; & x \geq b \end{cases} \quad (2)$$

Untuk menentukan derajat keanggotaan dengan menggunakan metode *max-min* menggunakan persamaan seperti di bawah [3]:

$$\mu_{df}(xi) = \max(\mu_{df}(xi), \mu_{kf}(xi)) \quad (3)$$

Sedangkan untuk metode defuzzifikasi menggunakan *Mean of Maximum* (MOM), yang dapat digambarkan oleh Gambar 5 [3], menggunakan persamaan 4:



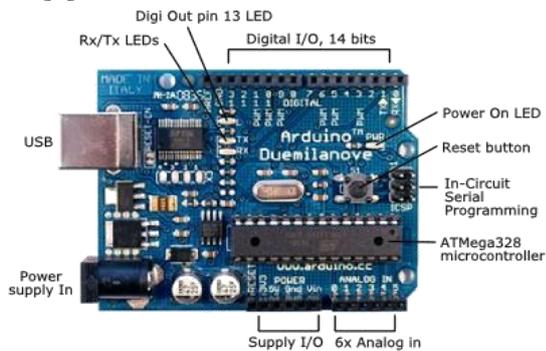
**Gambar 5.**Metode defuzzifikasi *Mean of Maximum* (MOM).

$$z^* = \frac{a+b}{2} \quad (4)$$

Pada metode ini, solusi *crisp* diperoleh dengan cara mengambil nilai rata-rata domain yang memiliki nilai keanggotaan maksimum.

## 2.2. Mikrokontroler

Modul mikrokontroler yang digunakan untuk rancangan ini adalah modul Arduino UNO dengan IC mikrokontroler Atmega 328. Gambar 6 menunjukkan susunan pin dari modul Arduino UNO [4].



**Gambar 6.** Modul Arduino UNO

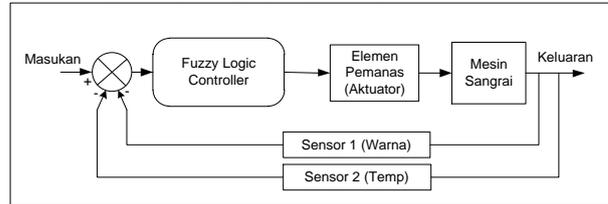
Spesifikasi Modul Arduino UNO adalah sebagai berikut:

- Jenis *Integrated Circuit* (IC): ATMEGA328
- Memori : 32 KBytes Flash dan 1024 EPROM
- I/O : 23 Programable I/O Lines
- *Speed Grade* : 0-20 MHz
- *Analog to Digital Converter* (ADC) : 8-channel, 10-bit
- 6 PWM channel

## 3. Perancangan Sistem.

### 3.1. Perangkat Keras

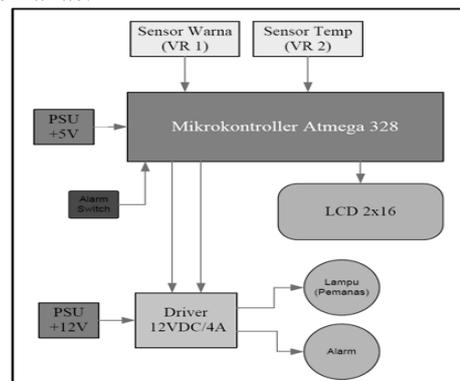
Rancangan sistem yang akan dibuat, ditunjukkan oleh Gambar 7. Sebuah pemanggang kopi dengan dua buah sensor dan sebuah elemen pemanas listrik. Masukan berupa nilai set poin tingkat kematangan biji kopi yang diinginkan.



**Gambar 7.** Diagram blok sistem pengendali elemen pemanas pemanggang kopi

Sensor pertama merupakan sensor warna yang akan mendeteksi tingkat kematangan biji kopi, sensor kedua adalah sensor temperatur yang akan mendeteksi nilai temperatur di dalam ruang sangrai. Nilai kedua sensor tersebut digunakan sebagai variabel masukan logika *fuzzy* yang akan menghasilkan variabel nilai keluaran berupa nilai *Pulse Width Modulation* (PWM) untuk menentukan besarnya daya listrik pada elemen pemanas.

Gambar 8 menunjukkan blok diagram sistem pengendali elemen pemanas pemanggang kopi dengan logika *fuzzy* menggunakan mikrokontroler. *Prototype* yang dibuat hanya akan mengilustrasikan prinsip kerja pengendali pemanas dari mesin *roaster*. Sensor warna dan sensor temperature akan digantikan oleh dua buah *variabel resistor* (VR), yaitu VR 1 dan VR 2. Sedangkan untuk pemanas listriknya menggunakan lampu pijar 25 watt yang dilengkapi *driver* 4A. Supply 5 VDC digunakan untuk catu daya mikrokontroler dan catu daya 12 VDC sebagai catu daya pemanas.



**Gambar 8.** Blok diagram sistem

*Liquid Crystal Display* (LCD) 2x16 digunakan untuk menampilkan nilai-nilai hasil perhitungan logika *fuzzy* yang dilakukan oleh mikrokontroler, yang nantinya akan dibandingkan dengan nilai

hasil simulasi di Matlab. Alarm berfungsi untuk memberikan informasi bahwa biji kopi yang disangrai telah mencapai tingkat kematangan tertentu. Tombol alarm digunakan untuk mematikan alarm secara manual.

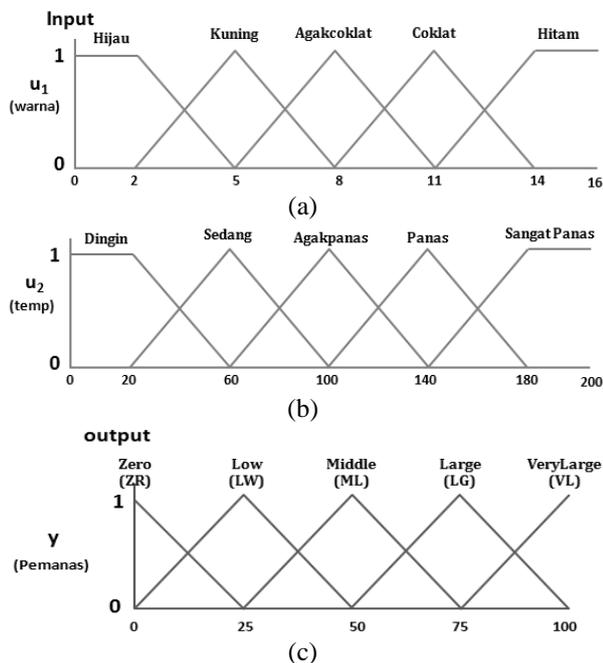
Desain *schematic* perangkat keras dapat dilihat pada Gambar 9. Perangkat keras terdiri dari modul mikrokontroler Arduino UNO, yang merupakan pengendali utama dari sistem *embedded* yang dibuat.

### 3.2. Logika fuzzy

Berdasarkan blok *Fuzzy Logic Controller* pada Gambar 2, maka:

Masukan :  $u_1$  (warna kopi (tingkat kematangan))  
 $u_2$  ( temperatur ruang sangrai( $^{\circ}C$ ))

Keluaran :  $y$  (pemanas(% dari nilai maksimum))  
 Terdapat tiga buah variabel data yang nantinya akan digunakan untuk membangun aturan-aturan *fuzzy*, yaitu dua data masukan dan satu data keluaran. Data masukan pertama berupa warna biji kopi, data masukan kedua temperatur ruang sangrai, sedangkan data keluaran berupa nilai elemen pemanas. Langkah pertama adalah menentukan fungsi keanggotaan atau *Membership Function* (MFs). Dalam rancangan ini kami menggunakan fungsi segitiga, baik untuk MFs masukan dan keluaran. Gambar 9 menunjukkan MFs masukan dan keluaran.



**Gambar 9.** (a) Fungsi keanggotaan masukan warna, (b) Fungsi keanggotaan masukan temperatur, (c) Fungsi keanggotaan keluaran pemanas.

Langkah selanjutnya adalah proses fuzzifikasi yaitu suatu proses untuk mengubah suatu masukan dari bentuk tegas (*crisp*) menjadi *fuzzy* (*variable*). Pada proses ini kami akan menentukan derajat keanggotaan untuk setiap nilai masukan.

Menentukan aturan-aturan *fuzzy* (*Fuzzy Rule Base*). Dalam perancangan ini aturannya ditentukan sebagai berikut:

- Antecedent* : warna ( $u_1$ ), temperatur ( $u_2$ )
- Consequence* : pemanas ( $y$ )
- Operator* : AND
- Banyaknya rule* : jumlah *input* = 2  
 jumlah MFs = 5, maka  
 jumlah rule =  $5^2 = 25$  rule

Tabel 1 menunjukkan aturan-aturan *fuzzy* yang digunakan dalam perancangan sistem logika *fuzzy* untuk kendali pemanas pemanggangkopi.

**Tabel 1.** Aturan-Aturan *Fuzzy*

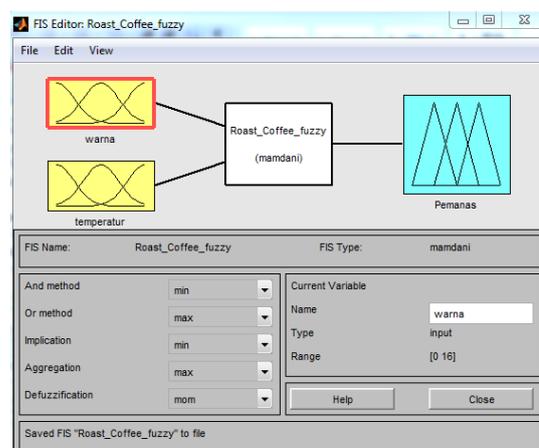
$U_1 \backslash U_2$	Hijau	Kuning	Agak Coklat	Coklat	Hitam
Dingin	VL	VL	LG	MD	ZR
Sedang	VL	VL	LG	MD	ZR
Agak Panas	VL	VL	LG	LW	ZR
Panas	VL	LG	MD	LW	ZR
Sangat Panas	VL	LG	MD	ZR	ZR

Di mana : VL = *very large*, LG = *large*, MD = *middle*, LW = *low*, dan ZR = *zero*.

Proses terakhir adalah *defuzzifikasi*, proses *defuzzifikasi* pada perancangan ini menggunakan persamaan 4. Nilai hasil *defuzzifikasi* ini akan diubah kedalam bentuk nilai PWM.

### 3.3. Simulasi dengan Matlab

*Toolbox FIS* (*Fuzzy Interference System*) pada Matlab digunakan untuk mensimulasikan rancangan logika *fuzzy* yang sudah dibuat. Gambar 10 menunjukkan *FIS Editor* dari sistem logika *fuzzy* pengendali pemanas pemanggang kopi.



**Gambar 10.** *FIS Editor* Sistem Logika *Fuzzy* Pengendali Pemanas Pemanggang Kopi.

Sedangkan Gambar 11 menunjukkan *rule viewer* yang mensimulasikan nilai masukan dan keluaran dengan aturan-aturan *fuzzy* yang dibuat. Data-data

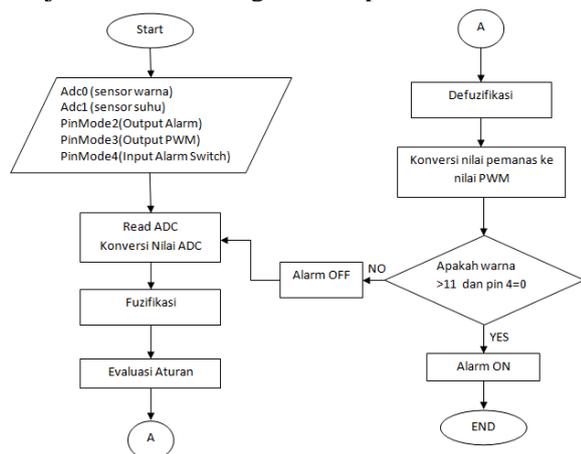
ini digunakan untuk memverifikasi hasil hitungan logika *fuzzy* yang dilakukan oleh mikrokontroler.



Gambar 11. Rule viewer

### 3.4 Aplikasi pada Mikrokontroler

Program mikrokontroler dibangun menggunakan bahasa C dengan algoritma seperti yang ditunjukkan dalam diagram alir pada Gambar 12.



Gambar 12. Diagram alir program Mikrokontroler

Proses pertama adalah inisialisasi pin-pin yang digunakan. Nilai ADC 0 diubah menjadi skala masukan warna, yaitu 1-16, ADC 1 diubah menjadi skala masukan temperatur yaitu 0-200. Kode program dalam bahasa C untuk proses ini adalah sebagai berikut:

```
adc0=analogRead(0);
adc1=analogRead(1);
warna=(adc0*16)/1023;
temp=(adc1*200)/1023;
```

Nilai-nilai masukan sensor selanjutnya diubah kedalam bentuk *MFs* dengan menggunakan kode program sebagai berikut:

```
//rumus membership "warna" (segitiga)
float MF_warna(float l, float b,
float c)
```

```
{ if (warna>=a&&warna<b)
  { member_warna=(warna-a)/(b-a); }
  if (warna>=b&&warna<c)
  { member_warna=(c-warna)/(c-b); }
  if (warna<2||warna>14)
  { member_warna=1; }
  if (warna>c||warna<a)
  { member_warna=0;}}
```

Sedang pemrograman evaluasi aturan dapat dilihat pada list di bawah:

```
//mencari nilai minimum (25 rule)
float fs[25]={0,1,2,3,4,5,6,7,8,9,10,
11,12,13,14,15,16,17,18,19,20,21,22,2
3,24}
{
if (hijau==1 && dingin==1)
{fs[0]=min(md_hijau,md_dingin);} else
{fs[0]=0;}
if (hijau==1 && sedang==1)
{fs[1]=min(md_hijau,md_sedang);} else
{fs[1]=0;}
if (hijau==1 && agakpanas==1)
{fs[2]=min(md_hijau,md_agakpanas);} else
{fs[2]=0;}
if (hijau==1 && panas==1)
{fs[3]=min(md_hijau,md_panas);} else
{fs[3]=0;}
if (hijau==1 && sangatpanas==1)
{fs[4]=min(md_hijau,md_sangatpanas);} else
{fs[4]=0;}
if (kuning==1 && dingin==1)
{fs[5]=min(md_kuning,md_sedang);} else
{fs[5]=0;}
if (kuning==1 && sedang==1)
{fs[6]=min(md_kuning,md_sedang);} else
{fs[6]=0;}
if (kuning==1 && agakpanas==1)
{fs[7]=min(md_kuning,md_sedang);} else
{fs[7]=0;}
if (kuning==1 && panas==1)
{fs[8]=min(md_kuning,md_sedang);} else
{fs[8]=0;}
if (kuning==1 && sangatpanas==1)
{fs[9]=min(md_kuning,md_sedang);} else
{fs[9]=0;}
```

Berdasar persamaan (4), proses defuzzifikasi dapat ditulis seperti kode program berikut:

```
/**PROSES DEFUZZYFIKASI MOM**
if
((st_zero==1)&&(st_low==0)&&(st_middl
e==0)&&(st_large==0)&&(st_verilarge==
0))
{ pmns=(zero)/2; }
```

```

if
((st_zero==0)&&(st_low==1)&&(st_middl
e==0)&&(st_large==0)&&(st_verilarge==
0))
{ pmns=(low_min+low_max)/2; }
if
((st_zero==1)&&(st_low==1)&&(st_middl
e==0)&&(st_large==1)&&(st_verilarge==
1))
{ pmns=(low_min+zero)/2; }
if
((st_zero==0)&&(st_low==0)&&(st_middl
e==1)&&(st_large==0)&&(st_verilarge==
0))
{ pmns=(middle_min+middle_max)/2; }

```

Setelah mendapatkan nilai defuzzifikasi, langkah terakhir adalah merubah nilai tersebut kedalam bentuk PWM. Kode programnya seperti dibawah ini.

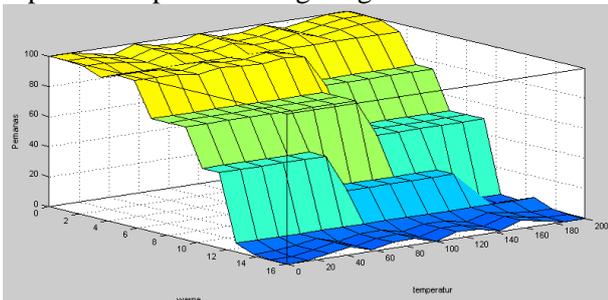
```

float pmns_pwm = (pmns/100)*255;
analogWrite (3, pmns_pwm);

```

#### 4. Hasil dan Analisa

Gambar 12 menunjukkan grafik respon elemen pemanas pemanggang kopi terhadap warna biji kopi dan temperatur ruang sangrai.



**Gambar 13.** Grafik respon pemanas pemanggang kopi terhadap warna biji kopi dan temperatur ruang sangrai

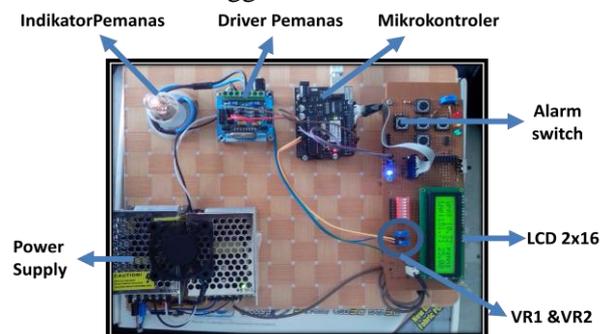
Dari Gambar 12 membuktikan bahwa logika *fuzzy* yang dibangun untuk sistem pengendalian elemen pemanas pemanggang kopi, sudah dapat dikatakan sesuai dengan keahlian seorang operator pada saat melakukan proses *roasting*. Sebagai contoh, saat biji kopi masih ditingkat kematangan 2 yang berarti warna biji kopi antara hijau dan kuning, sedangkan suhu ruang sangrai masih 40°C, maka tingkat pemanasan elemen pemanas harus 100%. Contoh lain saat warna biji kopi mendekati tingkat kematangan 16, berapapun suhu ruang sangrai pada saat itu, tingkat pemanasan elemen pemanas harus di bawah 5%. Karena pada tingkat kematangan ini proses *roasting* biji kopi harus segera diakhiri, agar biji kopi tidak rusak.

Gambar 13 menunjukkan *prototype* yang mengilustrasikan prinsip kerja pengendali elemen pemanas dari mesin pemanggang kopi. Mikrokontroler merupakan pengendali utama

sistem, dimana algoritma logika *fuzzy* ditanamkan ke dalam memori mikrokontroler tersebut.

*Port* masukan *analog* Arduino menggunakan tegangan 0-5V. *Port* masukan analog ini sudah terintegrasi dengan komponen ADC yang memiliki resolusi sebesar 10 bit. VR 1 akan memberikan nilai masukan ke ADC yang merepresentasikan nilai sensor warna dan VR 2 merepresentasikan nilai temperatur ruang sangrai. Masing-masing nilai ADC tersebut akan diubah menjadi skala nilai warna biji kopi dan temperature ruang sangrai. Sedangkan keluaran dari proses defuzzifikasi dikeluarkan oleh pin 3 yang didalamnya sudah terintegrasi dengan *Pulse Width Modulation* (PWM). Pin ini memiliki jangkauan *integer* 0-255, nilai ini digunakan untuk menggerakkan rangkaian driver pemanas dalam hal ini direpresentasikan oleh lampu pijar. Rangkaian *driver* akan merubah nilai PWM menjadi tegangan kerja lampu pijar yaitu 0 VDC sampai 12 VDC dengan keluaran arus maksimal 4 A.

Nilai-nilai masukan dan keluaran dapat dilihat pada tampilan LCD. Saat kita merubah-ubah nilai tahanan VR, secara *real time* mikrokontroler akan melakukan perhitungan logika *fuzzy* yang telah dibuat. Respon dari pemanas dapat langsung dilihat. Dalam sistem ini terdapat tambahan fitur alarm. *Alarm* akan aktif saat tingkat kematangan biji kopi bernilai > 15, ini untuk menandakan bahwa tingkat kematangan biji kopi mendekati batas maksimal. *Alarm* ini dapat dinonaktifkan secara manual menggunakan alarm *switch*.



**Gambar 14.** *Prototype* pengendali pemanas pemanggang kopidengan logika *fuzzy* menggunakan Mikrokontroler.

Perbandingan antara hasil simulasi Matlab dan perhitungan logika *fuzzy* yang dilakukan oleh mikrokontroler, dapat dilihat pada tabel 2.

**Tabel 2.** Hasil pengujian perhitungan logika *fuzzy* dengan Matlab dan Mikrokontroler.

NO	INPUT VALUE		OUTPUT VALUE		ERROR (%)
	WARNA (U <sub>1</sub> )	SUHU (U <sub>2</sub> )	MATLAB	HARDWARE	
1	0	0	100	100	0
2	2,08	20,14	100	99,67	0,33
3	5,08	28,54	97,5	97,33	0,17
4	7,99	49,27	75	75	0
5	9,23	113	75	75	0
6	10,01	125,5	25	25	0
7	10,56	64,13	50	50	0
8	11,03	113	25	25	0
9	12,89	153,27	4,5	4,64	0,14
10	13,75	200	1	1,05	0,05
11	16	200	0	0	0

Pada Tabel 2 dapat dilihat bahwa nilai keluaran yang dihasilkan simulasi Matlab dan perhitungan mikrokontroler, menunjukkan nilai yang relatif sama, dimana rata-rata nilai *error* hanya 0,06% atau mendekati 0 %. Ini membuktikan bahwa algoritma logika *fuzzy* untuk kendali pemanas pemanggang kopi, yang ditanam di sistem mikrokontroler dapat dikatakan berhasil.

Penerapan logika *fuzzy* pada mesin sangrai kopi menyebabkan prosedur kerja mesin ini berubah. Pengendalian tingkat elemen pemanas yang semula dikendalikan secara manual oleh seorang operator, dengan sistem ini, proses tersebut akan digantikan dengan sebuah sistem *embedded* menggunakan mikrokontroler sehingga prosesnya dapat berjalan otomatis.

Dari hasil pengujian perangkat keras, *prototype* yang dibuat dapat bekerja sesuai dengan rancangan yang dikehendaki. Untuk meningkatkan performa pengendalian, diperlukan tambahan *variable* masukan, seperti sensor temperature untuk pemanas dan variabel waktu *roasting*. Diperlukan pengujian untuk implementasi yang sebenarnya, yaitu mengganti VR dengan sensor-sensor yang sebenarnya dan mengganti lampu pijar dengan jenis elemen pemanas yang dipakai di pemanggang kopi

## 5. Kesimpulan

Dari hasil simulasi dan analisis data yang dihasilkan dapat diambil beberapa kesimpulan. Logika *fuzzy* yang dibangun untuk sistem pengendalian pemanas *roaster* kopi, sudah dapat dikatakan sesuai dengan keahlian seorang operator pada saat melakukan proses *roasting*. Algoritma Logika *fuzzy* yang ditanamkan ke mikrokontroler dapat dikatakan berhasil dengan

rata-rata error 0,06%. *Prototype* yang dibuat sudah bisa merepresentasikan sistem kerja pengendali pemanas *roaster* kopi yang sebenarnya dengan baik.

## Daftar Pustaka

- [1] Mulato, Sri. *Perancangan dan pengujian mesin sangrai biji kopi tipe silinder*. Pelita Perkebunan, 18, 31—45, 2002
- [2] <http://www.rumahkopi.com/2012/02/roasting-kopi.html>. diakses tanggal 24 November 2014
- [3] Suyanto. *Soft Computing Membangun Mesin Ber IQ Tinggi*. Penerbit Informatika. 2008.
- [4] McRoberts, Michael. *Beginning Arduino. Technology In Action*, 2010
- [5] Pudjo Widodo, Pabowo dan Rahmadya. *Penerapan Soft Computing dengan Matlab*. Rekayasa Sains. 2012

