

# PENGEMBANGAN PROGRAM LOGIKA FUZI PADA MIKROKONTROLER AT89C51

**Muhammad Arrofiq**

Program Diploma Teknik Elektro, Fakultas Teknik Universitas Gadjah Mada  
Jln. Sekip Selatan, Yogyakarta

Email: [rofiq@ugm.ac.id](mailto:rofiq@ugm.ac.id), [rofiqm@yahoo.com](mailto:rofiqm@yahoo.com)

## **Abstrak**

*Mikrokontroler tersusun atas mikroprosesor yang dilengkapi dengan pengingat, unit masukan/keluaran dan pewaktu, yang dikemas dalam satu chip. Mikrokontroler telah dirancang untuk mengendalikan sebuah proses tertentu. Tuntutan sebuah pengendali yang mengimplementasikan algoritma modern mendorong untuk memaksimalkan kemampuan mikrokontroler. Dengan kemampuan komputasi dan lebar data terbatas, mikrokontroler diharapkan mampu mengimplementasikan algoritma modern sederhana guna keperluan pengendalian. Penelitian ini mencoba merealisasikan program dengan algoritma fuji pada mikrokontroler keluarga AT8951. Fuzifikasi dilakukan dengan metoda melihat tabel, untuk meringankan beban komputasi mikrokontroler. Untuk mempermudah proses inferensi yang dilakukan mikrokontroler, aturan kendali ditetapkan menggunakan sebuah kode. Fungsi keanggotaan keluaran masih terbatas pada bentuk singleton. Defuzifikasi dilakukan dengan metoda center of gravity (COG). Dari hasil penelitian didapatkan terjadi adanya perbedaan hasil kecil antara perhitungan manual dengan perhitungan yang dilakukan mikrokontroler. Hal ini terjadi karena adanya pemotongan (truncating) pada sistem mikrokontroler. Waktu rata-rata penyelesaian algoritma fuji pada mikrokontroler adalah 1210 tick. Apabila sistem mikrokontroler menggunakan kristal 12 MHz, maka 1210 tick setara 1,2 mili detik.*

**Kata kunci:** Logika fuji, mikrokontroler AT8951

## **1. Pendahuluan**

Mikrokontroler, sebagai suatu terobosan teknologi mikroprosesor dan mikrokomputer, hadir memenuhi *market need* dan teknologi baru. Sebagai teknologi baru, yaitu teknologi semikonduktor dengan kandungan transistor yang lebih banyak namun hanya membutuhkan ruang yang kecil serta dapat diproduksi secara masal sehingga harganya menjadi lebih murah dibandingkan dengan mikroprosesor. Sebagai kebutuhan pasar, mikrokontroler hadir untuk memenuhi selera industri dan para konsumen akan kebutuhan dan keinginan alat bantu yang lebih baik dan canggih [1]. Salah satu jenis mikrokontroler yang umum, mudah didapat dan murah adalah AT8951, produksi perusahaan Atmel yang kompatibel dengan keluarga MCS-51.

Kebutuhan akan sebuah pengendali yang mengimplementasikan algoritma modern yang dikemas dalam ukuran kecil merupakan tuntutan lapangan. saat ini. Hal ini sudah diterapkan di peralatan-peralatan rumah tangga [2].

Penelitian ini bertujuan merealisasikan perangkat lunak algoritma fuji pada mikrokontroler AT8951 sesuai dengan kemampuan yang dimilikinya.

## 2. Logika Fuzi

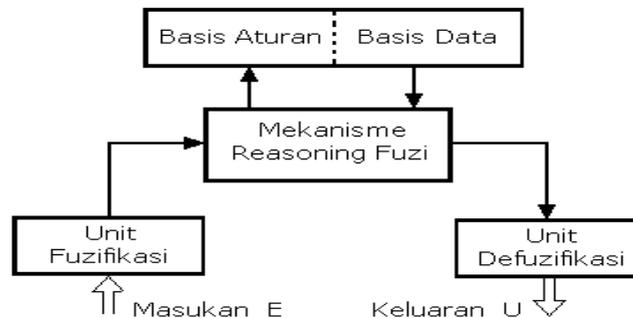
### 2.1 Susunan Dasar Pengendali Logika Fuzi

Pengendali logika fuzi secara tipikal menjadi satu pada sistem pengendali kalang tertutup. Gambar 1 menunjukkan diagram kotak sistem pengendali yang mengimplementasikan logika fuzi sebagai algoritma pengendalian.



Gambar 1. Sistem kendali logika fuzi tipikal

Algoritma logika fuzi memiliki beberapa tahapan seperti yang ditunjukkan Gambar 2.



Gambar 2. Susunan dasar pengendali logika fuzi

Gambar 2 menunjukkan elemen utama pengendali fuzi. Elemen-elemen utama pengendali fuzi adalah unit fuzifikasi, unit penarik kesimpulan, unit basis pengetahuan dan unit defuzifikasi). Jenis pengendali fuzi dapat berbeda, tergantung pada tujuan perancangan.

Basis pengetahuan fuzi berisi 2 informasi utama, yaitu basis data yang mendefinisikan fungsi keanggotaan fuzi yang digunakan sebagai nilai pada variabel sistem dan basis aturan yang memetakan nilai fuzi masukan menjadi nilai fuzi keluaran. Ada 2 jenis utama variable sistem, yaitu variabel masukan (E) yang didapatkan dari proses yang dikendalikan dan variabel keluaran yang digunakan oleh pengendali fuzi untuk mengendalikan proses. Untuk setiap variabel sistem yang digunakan dalam mengekspresikan aturan, nilai yang diijinkan harus didefinisikan pada semesta pembicaraan (*universe of discourse*). Pendefinisian hal tersebut merupakan langkah yang paling kritis pada proses perancangan dan dapat mempengaruhi unjuk kerja sistem. Nilai

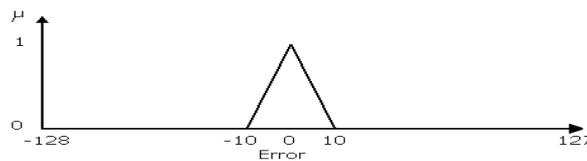
### 2.2. Program Fuzi pada Mikrokontroler Fuzifikasi

Proses fuzifikasi pada mikrokontroler dapat dilakukan dengan 2 cara, yaitu dengan perhitungan dan metoda melihat tabel (*look-up table*). Metode perhitungan dapat dilakukan apabila bentuk fungsi keanggotaannya sederhana. Kekangan metode ini adalah kemampuan komputasi mikrokontroler yang terbatas, lebar data mikrokontroler yang hanya 8-bit sehingga tidak mampu melaksanakan aritmatika yang kompleks. Metoda melihat tabel (*lookup table*) dapat digunakan untuk semua bentuk fungsi keanggotaan, karena nilai tingkat keanggotaan sudah didefinisikan dalam tabel. Proses fuzifikasi dilakukan dengan mengambil besarnya nilai tingkat keanggotaan pada tabel untuk sebuah masukan tertentu. Alamat tabel menunjukkan

besarnya masukan, isi tabel menunjukkan nilai tingkat keanggotaannya. Kekurangan metode ini adalah kebutuhan memori yang lebih banyak untuk menyimpan tabel.

Proses fuzifikasi yang digunakan saat ini adalah metode melihat tabel untuk kesederhanaan. Secara teori, nilai tingkat keanggotaan minimal 0 dan maksimal 1. Dalam metode tabel, nilai keanggotaan dibuat minimal 0 dan maksimal 255d. Gambar 3 menunjukkan contoh sebuah fungsi keanggotaan masukan error kelompok zero.

$$T(x; -10, 0, 10) = \begin{cases} 0 & , \quad x < 0 \\ (x+10)/10, & -10 \leq x \leq 0 \\ (10-x)/10, & 0 \leq x \leq 10 \\ 0 & , \quad x > 0 \end{cases} \quad (1)$$



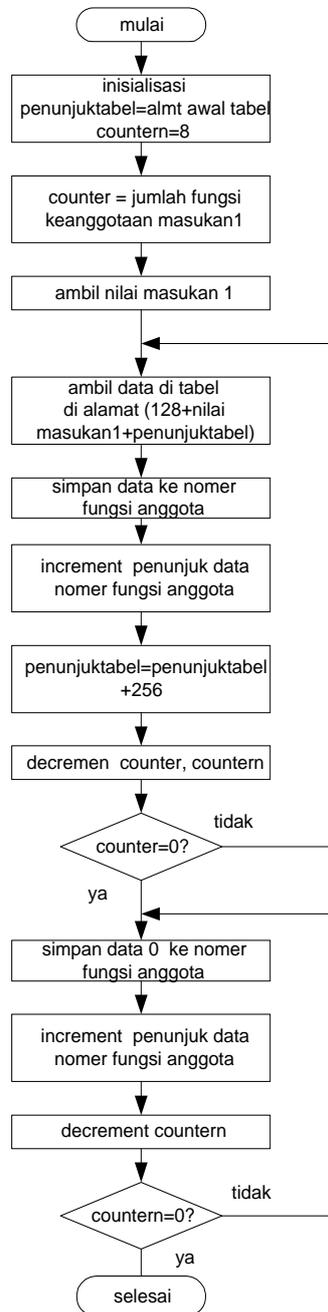
Gambar 3. Fungsi keanggotaan masukan error kelompok Zero

Nilai terendah dari masukan ditetapkan  $-128$  dan nilai tertinggi dari masukan ditetapkan  $127$ . Adapun jika terdapat masukan yang lebih rendah dari  $-128$  dapat diset sebagai nilai  $-128$ , begitu pula nilai masukan yang lebih tinggi dari  $127$  diset sebagai  $127$ . Dengan cara ini, jangkah masukan dapat ditangani dengan jangkah  $256$ . Besarnya tingkat keanggotaan pada tabel dihitung terlebih dahulu berdasarkan persamaan (1), disajikan pada Gambar 4 dengan nilai alamat awal tabel  $200h$ .

ALAMAT	DATA
200h	0
201h	0
.	0
.	0
.	0
273h	0
274h	0
275h	0
276h	0
277h	1A
278h	33
279h	4D
27Ah	66
27Bh	80
27Ch	99
27Dh	B3
27Eh	CC
27Fh	E6
280h	FF
281h	E6
282h	CC
283h	B3
284h	99
285h	80
286h	66
287h	4D
288h	33
289h	1A
28Ah	0
28Bh	0
.	0
.	0
.	0
2FFh	0

Gambar 4. Isi memori tabel

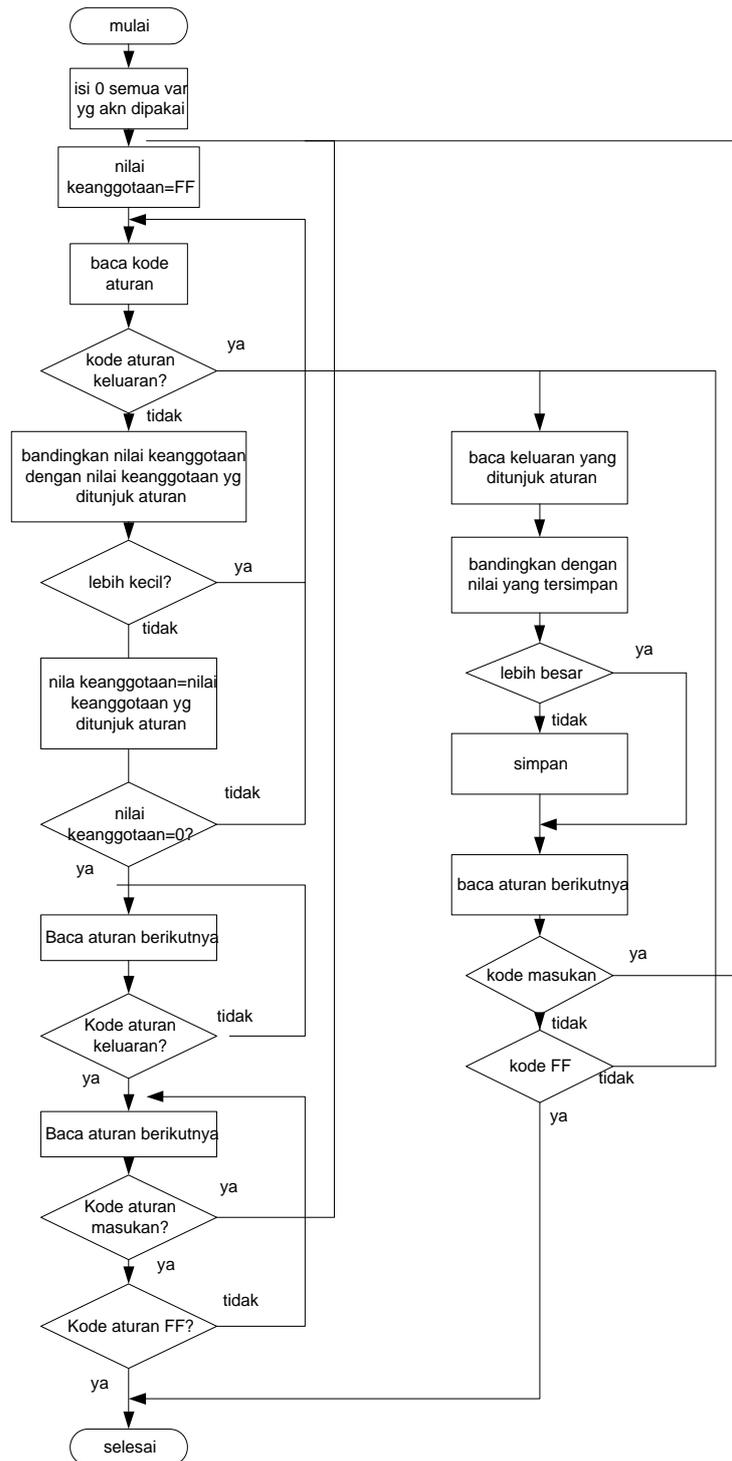
Berdasarkan Gambar 4, masukan Error -128 disimpan di alamat 200h. Dengan demikian, untuk nilai masukan error=0, tingkat keanggotaanya disimpan di alamat 280h dalam hal ini nilainya FFh. Program yang akan melakukan proses fuzifikasi memiliki diagram alir seperti ditunjukkan Gambar 5.



Gambar 5. Diagram alir proses fuzifikasi

### Evaluasi aturan

Aturan ditetapkan dengan sebuah format tertentu. Proses evaluasi aturan dilakukan dengan metoda MAXMIN. Diagram alir proses evaluasi aturan ditunjukkan Gambar 6.



Gambar 6. Diagram alir proses evaluasi aturan

**Defuzifikasi**

Proses defuzifikasi dilakukan dengan metode COG (*center of gravity*). Secara teoritis metoda ini membutuhkan komputasi/perhitungan yang rumit. Hal itu terjadi karena pengaruh bentuk fungsi keanggotaan keluaran. Perhitungan akan menjadi sederhana dan dapat dilakukan oleh mikroprosesor/mikrokontroler apabila bentuk fungsi keanggotaan keluaran berupa *singleton*. Persamaan proses defuzifikasi ditunjukkan persamaan (2).

$$COG = \frac{\sum_{i=1}^n \mu_i \cdot C_i}{\sum_{i=1}^n \mu_i} \quad (2)$$

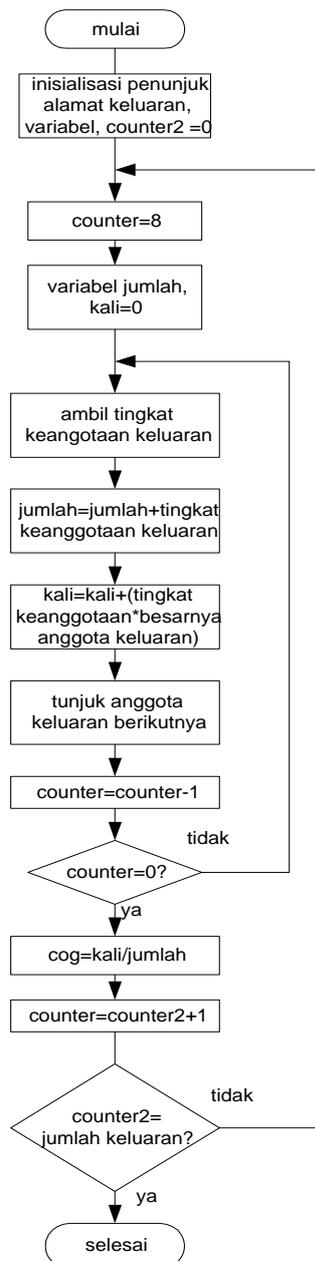
dengan :

$n$  = jumlah fungsi anggota keluaran

$\mu$  = tingkat keanggotaan

$C$  = anggota keanggotaan keluaran

Diagram alir proses defuzifikasi ditunjukkan Gambar 7.

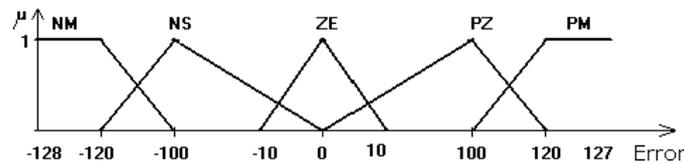


Gambar 7. Diagram alir proses defuzifikasi

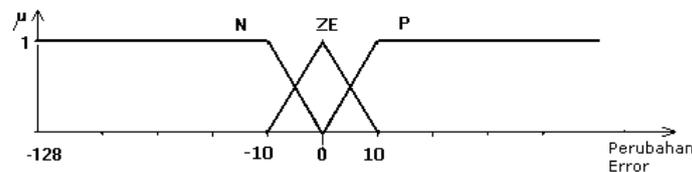
Variabel yang menyimpan jumlah perkalian antara tingkat keanggotaan dan anggota keluaran ( $\sum \mu_i.C_i$ ) memiliki lebar data 3 byte (24 bit). Variabel yang menyimpan jumlah tingkat keanggotaan ( $\sum \mu_i$ ) memiliki lebar data 2 byte (16-bit). Pada program proses defuzifikasi, proses pembagian antara ( $\sum \mu_i.C_i$ ) dengan ( $\sum \mu_i$ ) dapat dilakukan dengan menggunakan pembagian 2 byte (16-bit) atau pembagian 3 byte (24-bit). Pembagian 2 byte dapat dilakukan apabila nilai 1 byte tertinggi (bit ke16 sampai 23) dari ( $\sum \mu_i.C_i$ ) nilainya selalu 0. Nilai byte tertinggi selalu 0 atau tidak, dipengaruhi oleh aturan kendali yang dibuat. Apakah pada aturan kendali tersebut ada kemungkinan penjumlahan nilai besar melebihi 2 byte.

**Pengkodean Variabel Masukan, Keluaran dan Aturan**

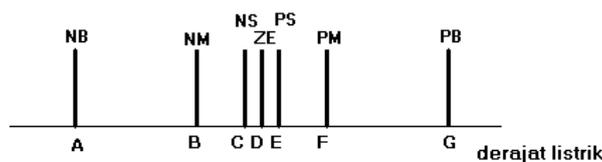
Setiap variabel masukan dibatasi maksimal memiliki 8 fungsi keanggotaan. Angka 8 dalam pemakaian sesungguhnya dirasakan lebih dari cukup. Sebagai contoh pengkodean, sebuah sistem pengendali fuzi memiliki 2 buah variabel masukan yaitu *error* (E) dan perubahan *error* (CE) 1 buah variabel keluaran (CI) dengan fungsi keanggotaan masing-masing ditunjukkan Gambar 8, 9, 10. Aturan pengendalian ditunjukkan Tabel 1.



Gambar 8. Fungsi keanggotaan masukan error



Gambar 9. Fungsi keanggotaan masukan perubahan error



Gambar 10. Fungsi keanggotaan keluaran

Tabel 1. Aturan kendali

		Perubahan Error		
		N	Z	P
E	NM	PB		
	NS	PM		
R	ZE	NS	Z	PS
	PS	NM		
R	PM	NB		

Format pengkodean variabel masukan disajikan Tabel 2.

Tabel 2. Format pengkodean masukan

Variabel Masukan		Kode (8 bit)			
Masukan	Fungsi Keanggotaan	Masukan/keluaran	x	No Var masukan	No fungsi dlm masukan
Error	NM	0	0	000	000
	NS	0	0	000	001
	ZE	0	0	000	010
	PS	0	0	000	011
	PM	0	0	000	100
Perubahan Error	N	0	0	001	000
	Z	0	0	001	001
	P	0	0	001	010

Berdasarkan Tabel 2, kode dibagi menjadi 3, yaitu masukan/keluaran, nomer variabel masukan dan nomer fungsi keanggotaan dalam masukan. Kode pada kolom masukan/keluaran adalah 0 (0 untuk masukan dan 1 untuk keluaran).

Berdasarkan Gambar 8 dan 9, terdapat 2 buah variabel masukan, yaitu masukan *error* dan masukan perubahan *error*. Masukan *error* diberi kode 0 (3 bit, 000) dan masukan perubahan *error* diberi kode 1 (3 bit, 001). Variabel masukan *error* memiliki 5 fungsi keanggotaan. Dari kiri ke kanan masing-masing NM, NS, ZE, PS, PM. Kode untuk masing-masing fungsi keanggotaan berturut-turut adalah 000, 001, 010, 011, 100. Variabel masukan perubahan *error* memiliki 3 fungsi keanggotaan, yaitu N, Z, P. Kode untuk masing-masing fungsi keanggotaan berturut-turut adalah 000, 001, 010.

Format pengkodean untuk variabel keluaran disajikan Tabel 3.

Tabel 3. Pengkodean keluaran

Keluaran	Kode (8 bit)		
Keluaran	Masukan/keluaran	x	No anggota dlm keluaran
NB	1	0000	000
NM	1	0000	001
NS	1	0000	010
ZE	1	0000	011
PS	1	0000	100
PM	1	0000	101
PB	1	0000	110

Berdasarkan Tabel 3, kode dibagi menjadi 2, yaitu masukan/keluaran dan nomer fungsi keanggotaan dalam variabel keluaran. Kode untuk variabel keluaran adalah 1 (0 untuk variabel masukan dan 1 untuk variable keluaran). Nomer fungsi keanggotaan dikodekan 3-bit. Pada Gambar 10, terdapat 7 fungsi anggota, yaitu NB, NM, NS, ZE, PS, PM, PB. Kode nomer anggota berturut-turut 000, 001, 010, 011, 100, 101, 110.

Format pengkodean aturan didasarkan pada skenario aturan pengendalian. Aturan kendali logika fuzi secara umum disajikan sebagai berikut:

**IF kondisi masukan 1 AND kondisi masukan 2  
THEN keluaran**

atau

**IF kondisi masukan 1 THEN keluaran**

atau

**IF kondisi masukan 2 THEN keluaran**

Format pengkodean aturan (dengan contoh di atas) ditetapkan sebagai berikut:

**Kode kondisi masukan 1**

Kode kondisi masukan 2

**Kode keluaran**

atau

**Kode kondisi 1****Kode keluaran**

atau

**Kode kondisi 2****Kode keluaran**

Skenario aturan kendali fuzi berdasarkan Tabel 1 memiliki kode yang ditunjukkan Tabel 4.

Tabel 4. Kode aturan kendali sesuai skenario

<b>Kode</b>	<b>Keterangan</b>
04h 80h	<b>IF error adalah PM THEN keluaran adalah NB</b>
03h 81h	<b>IF error adalah PS THEN keluaran adalah PS</b>
02h 08h 82h	<b>IF error adalah ZE AND perubahan error adalah N THEN keluaran adalah NS</b>
02h 09h 83h	<b>IF error adalah ZE AND perubahan error adalah Z THEN keluaran adalah ZE</b>
02h 0ah 84h	<b>IF error adalah ZE AND perubahan error adalah P THEN keluaran adalah NS</b>
01h 85h	<b>IF error adalah NS THEN keluaran adalah PM</b>
00h 86h	<b>IF error adalah NB THEN keluaran adalah PB</b>
FFh	<b>AKHIR ATURAN</b>

### 2.3. Pengujian Program

Pengujian program dilakukan dengan skema fuzi yang ditunjukkan Gambar 11.



01A5 F7	339	DB	247
01A6 F5	340	DB	245
01A7 F2	341	DB	242
01A8 F0	342	DB	240
01A9 ED	343	DB	237
01AA EB	344	DB	235
01AB E8	345	DB	232
01AC E6	346	DB	230
01AD E3	347	DB	227
01AE E0	348	DB	224
01AF DE	349	DB	222
01B0 DB	350	DB	219
01B1 D9	351	DB	217
01B2 D6	352	DB	214
01B3 D4	353	DB	212
01B4 D1	354	DB	209
01B5 CF	355	DB	207
01B6 CC	356	DB	204
01B7 C9	357	DB	201
01B8 C7	358	DB	199
01B9 C4	359	DB	196
01BA C2	360	DB	194
01BB BF	361	DB	191
01BC BD	362	DB	189
01BD BA	363	DB	186
01BE B8	364	DB	184
01BF B5	365	DB	181
01C0 B3	366	DB	179
01C1 B0	367	DB	176
01C2 AD	368	DB	173
01C3 AB	369	DB	171
01C4 A8	370	DB	168
01C5 A6	371	DB	166
01C6 A3	372	DB	163
01C7 A1	373	DB	161
01C8 9E	374	DB	158
01C9 9C	375	DB	156
01CA 99	376	DB	153
01CB 96	377	DB	150
01CC 94	378	DB	148
01CD 91	379	DB	145
01CE 8F	380	DB	143
01CF 8C	381	DB	140
01D0 8A	382	DB	138
01D1 87	383	DB	135
01D2 85	384	DB	133
01D3 82	385	DB	130
01D4 80	386	DB	128
01D5 7D	387	DB	125
01D6 7A	388	DB	122
01D7 78	389	DB	120
01D8 75	390	DB	117
01D9 73	391	DB	115
01DA 70	392	DB	112
01DB 6E	393	DB	110
01DC 6B	394	DB	107
01DD 69	395	DB	105
01DE 66	396	DB	102
01DF 63	397	DB	99
01E0 61	398	DB	97
01E1 5E	399	DB	94
01E2 5C	400	DB	92
01E3 59	401	DB	89
01E4 57	402	DB	87

01E5 54	403	DB 84
01E6 52	404	DB 82
01E7 4F	405	DB 79
01E8 4D	406	DB 77
01E9 4A	407	DB 74
01EA 47	408	DB 71
01EB 45	409	DB 69
01EC 42	410	DB 66
01ED 40	411	DB 64
01EE 3D	412	DB 61
01EF 3B	413	DB 59
01F0 38	414	DB 56
01F1 36	415	DB 54
01F2 33	416	DB 51
01F3 30	417	DB 48
01F4 2E	418	DB 46
01F5 2B	419	DB 43
01F6 29	420	DB 41
01F7 26	421	DB 38
01F8 24	422	DB 36
01F9 21	423	DB 33
01FA 1F	424	DB 31
01FB 1C	425	DB 28
01FC 1A	426	DB 26
01FD 17	427	DB 23
01FE 14	428	DB 20
01FF 12	429	DB 18
0200 0F	430	DB 15
0201 0D	431	DB 13
0202 0A	432	DB 10
0203 08	433	DB 8
0204 05	434	DB 5
0205 03	435	DB 3
0206 00	436	DB 0
0207 00	437	DB 0
.	.	.
.	.	.
.	.	.
0285 00	563	DB 0

Pengujian program logika fuzzy pada mikrokontroler dengan berbagai variasi masukan disajikan Tabel 5.

Tabel 5. Nilai hasil algoritma fuzzy dengan beberapa variasi masukan

		<i>CHANGE OF ERROR</i>											
		-60	-50	-45	-25	-15	0	15	25	45	50	60	
<i>ERROR</i>	-125	-100	-100	-100	-100	-100	-100	-100	-100	-100	-100	-100	-100
	-100	-100	-100	-100	-100	-100	-100	-100	-100	-100	-100	-100	-100
	-75	-82	-82	-74	-65	-65	-75	-55	-55	-63	-69	-69	-69
	-50	-63	-63	-57	-42	-45	-50	-33	-25	-35	-38	-38	-38
	-25	-44	-44	-40	-30	-27	-26	-14	-10	-6	-7	-7	-7
	0	-25	-25	-23	-13	-8	0	7	12	22	25	25	25
	25	6	6	5	10	13	25	26	30	39	43	43	43
	50	37	37	34	25	32	50	44	41	56	62	62	62
	75	68	68	62	54	54	74	64	64	73	81	81	81
	100	100	100	100	100	100	100	100	100	100	100	100	100
125	100	100	100	100	100	100	100	100	100	100	100	100	

Dengan menggunakan program Emulator 8051 TS Control dengan frekuensi kristal 12 MHz proses komputasi fuzi diselesaikan rata-rata 1210 tick atau setara dengan 1,21 mili detik.

### 3. KESIMPULAN

Dari hasil penelitian dapat disimpulkan:

1. Proses fuzifikasi menggunakan metoda *look-up table* meringankan beban komputasi pada mikrokontroler.
2. Dengan menggunakan metoda *look-up table* pada proses fuzifikasi, bentuk fungsi keanggotaan masukan yang dapat dipakai tidak terbatas pada segitiga dan trapesium.
3. Waktu rata-rata proses algoritma fuzi adalah 1210 ticks.

### DAFTAR PUSTAKA

- [ 1] Putra, A., F., 2002, "**Belajar Mikrokontroler AT89C51/52/55: Teori dan Aplikasi**" Gava Media, Yogyakarta.
- [ 2] Chen, C.H., 1996. "**Fuzzy Logic and Neural Network Handbook**" McGraw-Hill Inc.
- [ 3] Arrofiq, M., 2004, "**Pengaturan Kecepatan Motor DC secara Fuzi berbasis Mikrokontroler M68HC11**" Tesis Program S2, Program Pascasarjana Universitas Gadjah Mada, Yogyakarta.
- [ 4] Jamshidi, M., Vadiie, N, Ross, J.T., 1993, "**Fuzzy Logic and Control: Software and Hardware Applications**" Prentice-Hall International, Inc.
- [ 5] Malik, M.I, dan Anistardi, 1999, "**Bereksperimen dengan Mikrokontroler 8031**" Elexmedia Komputindo, Jakarta.
- [ 6] Motorola, 1991, "**M68HC11 Reference Manual Rev.3**" Motorola Inc.
- [ 7] Widodo, T.S., 2005, "**Sistem Neuro Fuzzy untuk Pengolahan Informasi, pemodelan , dan Kendali**" Edisi Pertama, Graha Ilmu, Yogyakarta
- [ 8] Yan, Jun, M. Ryan, J. Power , 1994, "**Using Fuzzy Logic**" Prentice Hall International
- [ 9] ....., "**Fuzzy logic engine for the 68HC11**"  
<http://www.programmersheaven.com/zone5/cat26/1288.htm>