

## Makalah Seminar Tugas Akhir

### Pengaturan Kecepatan Motor DC Dengan *Adaptive Fuzzy Logic Controller* Metode *Tuning Output*

Peter Christianto<sup>[1]</sup>, Darjat, S.T, M.T<sup>[2]</sup>, Iwan Setiawan, S.T, M.T<sup>[2]</sup>

Jurusan Teknik Elektro, Fakultas Teknik, Universitas Diponegoro  
Jln. Prof. Sudharto, Tembalang, Semarang, Indonesia

#### Abstrak

Tugas Akhir ini bertujuan untuk mengimplementasikan kendali logika fuzzy adaptif pada motor DC. Kendali logika fuzzy adaptif metode tuning output yang digunakan merupakan sistem kendali modern yang mudah karena tidak perlu mencari model matematis dari sistem, tetapi tetap efektif karena memiliki respon sistem yang stabil.

Dalam tugas akhir ini dibuat alat pengatur kecepatan motor DC yang dikontrol dengan menggunakan kendali logika fuzzy dan fuzzy adaptif metode tuning output. Kedua sistem kendali memiliki fungsi keanggotaan yang sama untuk input dan outputnya, selain itu basis aturan yang digunakan juga sama. Penggerak motor menggunakan metode PWM (Pulse Width Modulation). Input setting point diberikan melalui keypad yang akan diproses oleh mikrokontroler. Hasil keluaran berupa kecepatan putaran motor DC akan disensor oleh optointerruptor yang akan digunakan sebagai umpan-balik dalam sistem kendali.

Pengujian sistem dilakukan dengan variasi setting point terhadap kedua sistem kendali. Hasil penelitian dengan membandingkan kedua sistem kendali didapatkan bahwa kendali logika fuzzy adaptif metode tuning output menghasilkan respon sistem yang lebih baik daripada kendali logika fuzzy.

**Kata kunci** : Kendali Logika Fuzzy, Fuzzy adaptif, Tuning output

## I. PENDAHULUAN

### 1.1 Latar Belakang

Dewasa ini dengan semakin berkembangnya dunia industri, kebutuhan akan sistem kendali yang efisien dan mampu meningkatkan kualitas dan hasil industri menjadi sedemikian penting. Untuk itu perlu dikembangkan suatu sistem kendali yang mampu mengatur sistem secara otomatis untuk menghasilkan hasil produksi seperti yang diinginkan sehingga dapat mengurangi error. Penggunaan logika fuzzy dalam sistem kendali sudah bukan merupakan hal yang baru, sudah begitu banyak aplikasi-aplikasi sistem kendali yang menggunakan teori logika fuzzy yang dihasilkan.

Kemajuan penggunaan logika fuzzy dalam sistem kendali ini diiringi dengan perkembangan teknik sistem kendali logika fuzzy, salah satunya adalah logika fuzzy adaptif. Pada penggunaan sistem kendali logika fuzzy ini diperlukan pada aplikasi dengan lingkungan yang selalu berubah-ubah, akan cukup menyulitkan bagi pembuat sistem kendali untuk mendapatkan hasil kendali yang tepat dan memuaskan. Karena pembuat sistem kendali perlu mengubah dan menyesuaikan parameter kendali seperti *fuzzy rules*, *membership function* dan lain sebagainya pada lingkungan yang berubah-ubah. Sistem kendali logika fuzzy adaptif akan mengubah dan menyesuaikan sistem kendali secara otomatis sesuai dengan kelakuan sistem yang dikehendaki. Salah satu teknik

kendali logika fuzzy adaptif adalah dengan metode tuning output. Metode tuning output akan mengamati perubahan kesalahan (error) untuk melakukan proses adaptifnya. Karena penggunaan sistem kendali logika fuzzy adaptif yang begitu menguntungkan disini penulis mencoba untuk mempelajari salah satu metode sistem kendali logika fuzzy adaptif yaitu tuning output pada aplikasi pengatur kecepatan motor DC.

### 1.2 Tujuan

Tujuan dari penelitian tugas akhir ini adalah mempelajari sistem kendali logika fuzzy adaptif metode tuning output yang diaplikasikan pada perangkat pengatur kecepatan motor DC yang mampu mencapai kecepatan putaran sesuai dengan *Setting Point* (SP) yang diberikan.

### 1.3 Pembatasan Masalah

Pembatasan masalah pada Tugas Akhir ini meliputi:

- Mikrokontroler yang digunakan adalah mikrokontroler AVR ATmega8535.
- Bahasa pemrograman yang digunakan adalah bahasa C yang diadaptasikan pada *software Code Vision AVR*.
- Komputer (PC) hanya digunakan untuk mengetahui respon transient motor DC dengan menggunakan program Delphi.
- Range masukan setting point 0–1300rpm.
- Teknik kendali yang digunakan adalah kendali logika fuzzy dan fuzzy adaptif metode

<sup>1</sup> Mahasiswa Jurusan Teknik Elektro UNDIP

<sup>2</sup> Staf Pengajar Jurusan Teknik Elektro UNDIP

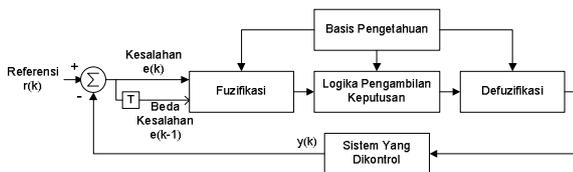
tuning output dengan dua fungsi keanggotaan segitiga dan trapesium untuk masukan dan satu fungsi keanggotaan singleton untuk keluaran.

- Metode pengambilan keputusan menggunakan metode Max-Min.
- Metode defuzzifikasi yang digunakan adalah metode COG (*Centre of Gravity*).
- Pembuatan perangkat tidak memuat pembebanan motor DC dan gangguan terhadap motor DC.
- Tidak membahas mekanik motor.

## II. DASAR TEORI

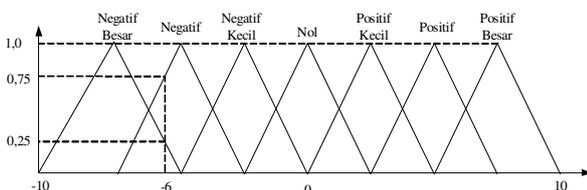
### 2.1 Kendali Logika Fuzzy

*Fuzzy* berarti samar, kabur atau tidak jelas. *Fuzzy* adalah istilah yang dipakai oleh Lotfi A Zadeh pada bulan Juli 1964 untuk menyatakan kelompok / himpunan yang dapat dibedakan dengan himpunan lain berdasarkan derajat keanggotaan dengan batasan yang tidak begitu jelas (samar), tidak seperti himpunan klasik yang membedakan keanggotaan himpunan menjadi dua, himpunan anggota atau bukan anggota.



Gambar 2.1 Mekanisme kendali logika fuzzy kalang tertutup

Kendali logika fuzzy dilakukan dalam tiga tahap, yaitu fuzzifikasi, evaluasi aturan dan defuzzifikasi. Komponen fuzzifikasi berfungsi untuk memetakan masukan data tegas ke dalam himpunan fuzzy menjadi nilai fuzzy dari beberapa variabel linguistik masukan.



Gambar 2.2 Proses fuzzifikasi

Basis pengetahuan berisi pengetahuan sistem kendali sebagai pedoman evaluasi keadaan sistem untuk mendapatkan keluaran kendali sesuai yang diinginkan perancang. Basis pengetahuan terdiri dari basis data dan basis aturan fuzzy.

#### 1. Basis Data

Basis data merupakan komponen untuk mendefinisikan himpunan fuzzy dari masukan dan keluaran.

#### 2. Basis Aturan Fuzzy

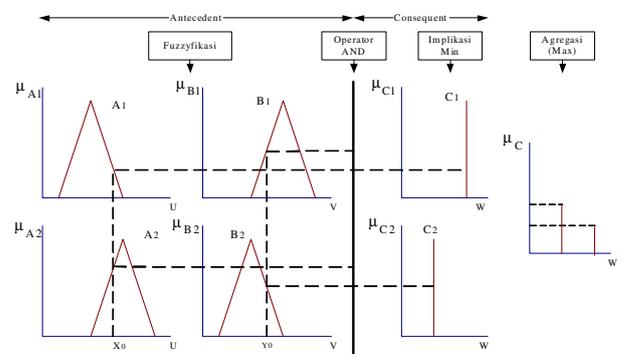
Basis Aturan fuzzy merupakan kumpulan pernyataan aturan 'IF-THEN' yang didasarkan kepada pengetahuan pakar.

Metode yang digunakan dalam penentuan basis aturan fuzzy adalah menggunakan metode *trial and error*.

Logika pengambilan keputusan disusun dengan cara menuliskan aturan yang menghubungkan antara masukan dan keluaran sistem fuzzy. Aturan ini diekspresikan dalam kalimat: 'jika <masukan> maka <keluaran>'.

Metode ini mempunyai bentuk aturan seperti persamaan 2.1.

$$\text{IF } x \text{ is } A \text{ and } y \text{ is } B \text{ then } z = k \quad (2.1)$$



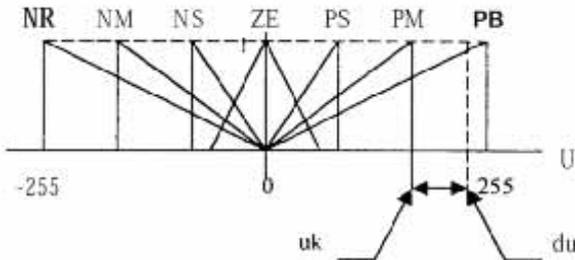
Gambar 2.3 Proses pengambilan keputusan metode Sugeno

Defuzzifikasi dapat didefinisikan sebagai proses perubahan besaran fuzzy yang disajikan dalam bentuk himpunan-himpunan fuzzy keluaran dengan fungsi keanggotaan untuk mendapatkan kembali bentuk tegasnya (*crisp*).

### 2.2 Kendali Logika Fuzzy Adaptif Metode Tuning Output

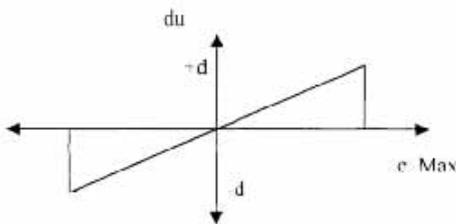
Ada berbagai jenis metode *adaptive fuzzy logic*, pada Tugas akhir ini digunakan metode tuning output, dimana metode tuning output yang digunakan merupakan modifikasi dari metode tuning *membership function* pada variabel output. Pada metode tuning *membership function* pada variabel output, apabila sejumlah fuzzy set pada input kesalahan (error) dan selisih kesalahan ( $\Delta$  error) didefinisikan dengan *membership function* bentuk segitiga tertentu, maka jumlah fuzzy set yang sama akan didefinisikan untuk variabel output ( $U(t)$ ), tetapi *membership function* dari variabel output tersebut dapat disesuaikan dalam semesta pembicaraan output. Untuk *membership function*  $\mu_k(u)$  dari fuzzy set k dinyatakan dengan bentuk garis lurus, untuk itu persamaan pengaturan dapat diberikan sebagai berikut:

$$\mu_k(u) = \frac{1}{uk + du} u \quad (2.2)$$



Gambar 2.4 Fuzzy set untuk output dengan variabel membership

Disini  $\mu_k(u) = 1$  sebelum dilakukan penyesuaian terhadap *membership function*. *Membership function* dari fuzzy set dinyatakan dalam semesta pembicaraan output dapat dengan mudah dituning dengan mengubah nilai  $du$ , salah satunya dengan mengamati perubahan kesalahan (error), dimana perubahan nilai  $du$  sebanding dengan besarnya perubahan nilai error.



Gambar 2.5 Grafik perubahan  $du$  terhadap error

Perubahan nilai  $du$  akan meningkatkan kontribusi dari tiap rule untuk error yang besar atau memperkecil output kontrol dengan mengurangi kontribusi dari tiap rule untuk error yang kecil.

Pada proses adaptif dengan metode tuning output, tidak dilakukan penggeseran *membership function* output yang digeser lebih besar atau lebih kecil dibanding dengan error yang terjadi untuk proses adaptifnya, tetapi nilai outputnya yang digeser dengan mengamati error yang terjadi, sehingga seolah-olah *membership function* outputnya juga tergeser.

### 2.3 Pengaturan Kecepatan pada Motor DC

Pengaturan kecepatan pada motor dc dapat dilakukan dengan mengatur besar tegangan masukan. Kecepatan putar motor dc (N) dapat dirumuskan dengan persamaan di bawah ini :

$$N = \frac{V_{TM} - I_A R_A}{K\Phi} \quad (2.3)$$

$V_{TM}$  merupakan tegangan terminal,  $I_A$  adalah arus jangkar motor,  $R_A$  adalah hambatan jangkar motor,  $K$  adalah konstanta motor, dan  $\Phi$  merupakan fluks magnet yang terbentuk pada motor.

Kecepatan motor dc berbanding lurus dengan suplai tegangan, sehingga pengurangan suplai tegangan akan menurunkan kecepatan motor dan penambahan suplai tegangan akan menambah kecepatan motor. Cara ini dilakukan dengan mengatur tegangan *power supply* untuk mendapatkan kecepatan motor yang dikehendaki.

### 2.4 Definisi dan Spesifikasi Respon Transient Sistem

Respon transient merupakan respon sistem dalam domain waktu yang sangat penting dalam studi tentang sistem.

#### A. Rise Time ( $t_r$ )

Untuk sistem tipe *underdamped*, *rise time* didefinisikan sebagai waktu yang dibutuhkan oleh respon output untuk bergerak dari 0 % hingga 100% terhadap kondisi mantapnya. Sedangkan untuk sistem tipe *overdamped*, lebih sering didefinisikan sebagai waktu yang bergerak dari 10 % hingga 90% relatif terhadap kondisi mantapnya.

#### B. Delay time ( $t_d$ )

Adalah waktu yang dibutuhkan oleh respons output untuk mencapai 50 % dari kondisi mantapnya.

#### C. Peak Time ( $t_p$ )

Adalah waktu yang dibutuhkan oleh respon output untuk yang pertama kali mencapai tinggi maksimumnya.

#### D. Maximum overshoot ( $M_p$ )

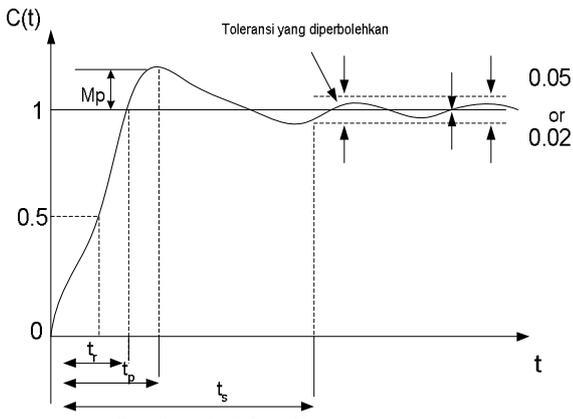
Adalah tinggi maksimum (puncak tertinggi) yang diukur dari kondisi mantapnya. Sering pula didefinisikan parameter *maximum percent overshoot*, sebagai :

$$\frac{y(t_p) - y(t \rightarrow \infty)}{y(t_p)} \times 100\% \quad (2.4)$$

*Maximum overshoot* terjadi pada saat *peak time*.

#### E. Settling time ( $t_s$ )

Adalah waktu yang dibutuhkan oleh respon output untuk mencapai kondisi mantapnya dengan toleransi yang diberikan. Dalam praktek toleransi yang diberikan untuk kondisi yang mantap dapat berkisar  $\pm 0,05\%$ . Pada beberapa aplikasi (sintesa) dapat diijinkan toleransi  $\pm 2\%$  hingga  $\pm 5\%$ .

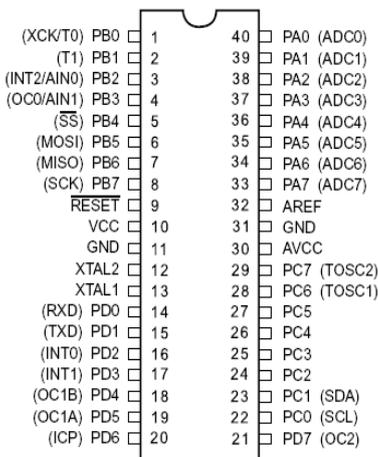


Gambar 2.6 Grafik respon transient sistem

Respon transient sistem sering dipelajari berkaitan dengan untuk kerja hasil sintesa yang dilakukan. Misalkan pada sistem kontrol, ingin diketahui hasil rancangan suatu kontroler terhadap sistem yang dikontrol, respon transient menjadi sangat penting artinya.

### 2.5 Mikrokontroler AVR ATMEGA8535

Mikrokontroler AVR ATMEGA8535 telah didukung penuh dengan program dan sarana pengembangan seperti: kompilator-kompilator C, simulator program, emulator dalam rangkaian, dan kit evaluasi.



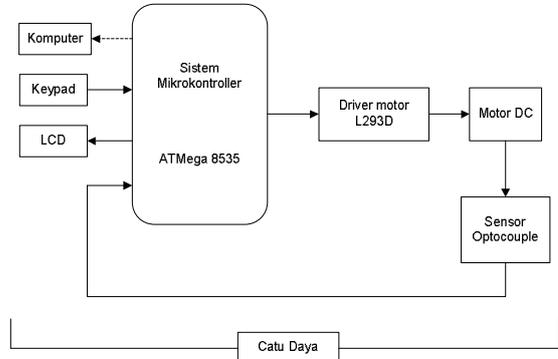
Gambar 2.7 Konfigurasi pin-pin ATMEGA8535

Mikrokontroler AVR ATMEGA8535 memiliki ruang pengalamatan memori data dan memori program terpisah. Memori program yang terletak dalam *Flash PEROM* tersusun dalam word atau 2 byte karena setiap instruksi memiliki lebar 16-bit atau 32-bit. Selain itu, mikrokontroler ini memiliki *4KByteX16-bit Flash PEROM* dan memiliki 12-bit *Program Counter (PC)* sehingga mampu mengamati isi *Flash*.

## III. PERANCANGAN

### 3.1 Perancangan Perangkat Keras

Perancangan perangkat keras pada perangkat pengaturan kecepatan motor DC ini meliputi mikrokontroler AVR ATmega 8535, sensor *optocoupler*, *Keypad*, LCD, komputer (PC), *driver* motor dc dan rangkaian catu daya. Secara umum perancangan yang akan dibuat dapat dijelaskan pada gambar 3.1 di bawah ini.

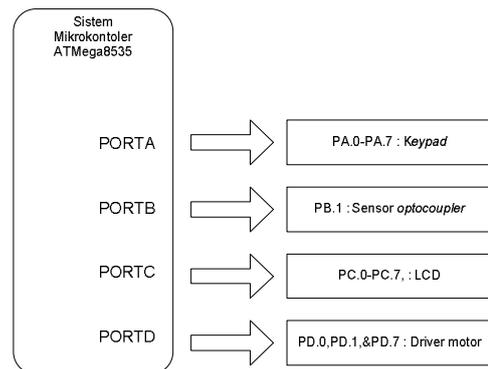


Gambar 3.1 Blok diagram sistem

#### 3.1.1 Sistem Minimum AVR ATMEGA8535

Mikrokontroler AVR ATMEGA8535 yang digunakan sudah mendukung kemampuan *In-System Programming*, yaitu pengisian program ke dalam sistem dengan mikrokontroler yang sedang digunakan bisa dilakukan.

Mikrokontroler AVR ATMEGA8535 berfungsi sebagai pengendali utama, yaitu sebagai penerima data masukan, melakukan pengendalian perangkat dengan menggunakan algoritma kendali logika *fuzzy* dan *fuzzy* adaptif metode tuning output, dan menghasilkan keluaran berupa PWM yang digunakan untuk melakukan aksi kendali terhadap motor DC melalui driver motor L293D.



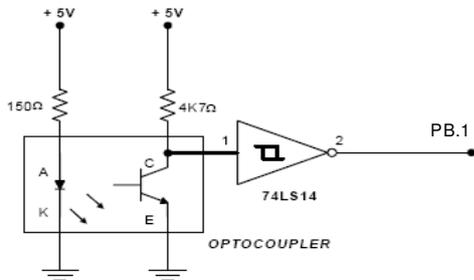
Gambar 3.2 Alokasi port pada sistem minimum mikrokontroler ATMEGA8535

Fitur-fitur dari mikrokontroler yang digunakan meliputi *timer 0* yang digunakan untuk menghasilkan interupsi, *timer 1* yang difungsikan sebagai *counter* yang akan menghitung jumlah pulsa yang dihasilkan oleh sensor optocoupler dan *timer 2* yang difungsikan sebagai PWM mode *phase correct* untuk membangkitkan sinyal PWM sebagai sinyal

kontrol, sedangkan PORTA dan PORTC digunakan sebagai digital I/O biasa.

### 3.1.2 Sensor Optocoupler

Sensor optocoupler digunakan untuk menghitung kecepatan dari motor DC. Rangkaian sensor motor DC seperti pada gambar 3.11. Data hasil sensor dihubungkan ke port B.1 mikrokontroler untuk diproses lebih lanjut.



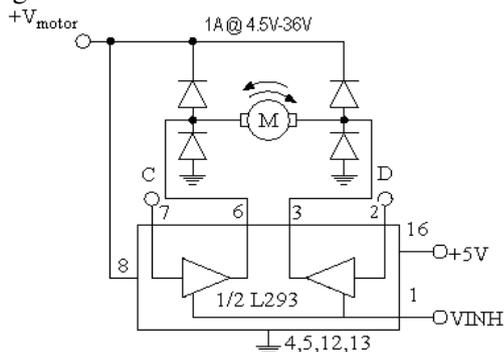
Gambar 3.3 Rangkaian sensor optocoupler

Dengan mengetahui prinsip dari optocoupler ini maka harus dibuat penghalang antara transistor dan LED, pada penghalang itu diberi lubang. Ketika berputar output dari transistor akan mengalami high dan low. Di sini penulis membuat 40 lubang pada kipas motor untuk mempermudah perhitungan. Agar output dari rangkaian ini menjadi lebih akurat maka ditambahkan schmitt trigger (74LS14), yang berlaku sebagai gerbang logika NOT. Jadi ketika transistor dan LED terdapat halangan maka keluaran sensor yang masuk ke mikrokontroler akan berlogika low (0) dan sebaliknya ketika transistor dan LED tidak terdapat halangan maka keluaran sensor akan berlogika high (1).

### 3.1.3 Driver Motor L293D

Pada Tugas Akhir ini digunakan driver motor IC L293D yang memungkinkan untuk menjalankan motor dengan arus hingga 600mA per channel. Driver motor ini dilengkapi dengan dioda internal yang berfungsi untuk mencegah kerusakan IC dari tegangan balik yang dibangkitkan oleh motor DC.

Gambar dari rangkaian driver ini adalah sebagai berikut :



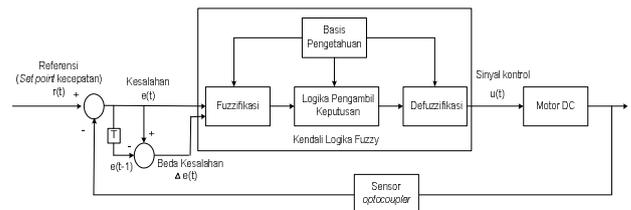
Gambar 3.4 Rangkaian Driver motor L293D

## 3.2 Perancangan Perangkat Lunak (Software)

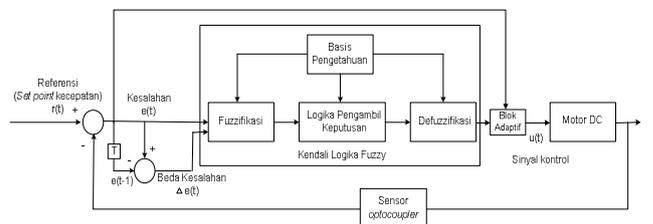
Pada Tugas Akhir ini, untuk perancangan perangkat lunak digunakan bahasa C dengan kompilator Code Vision AVR 1.24 versi standar. Pada bagian pemrograman digunakan bahasa C karena kemudahan, kesederhanaan, dan fleksibilitasnya.

Perangkat yang dibuat pada Tugas Akhir ini adalah perangkat yang akan mengatur kecepatan motor DC sesuai dengan referensi kecepatan yang diberikan. Perangkat ini dikendalikan menggunakan 2 sistem kendali, yaitu kendali logika fuzzy dan fuzzy adaptif metode tuning output. Masukan pengendali berupa error jarak dan perubahan error jarak. Keluaran dari pengendali logika fuzzy ini berupa sinyal kontrol yang kemudian mengendalikan plant motor dc.

Blok diagram pengendalian secara umum dapat dilihat pada Gambar 3.9 dan gambar 3.10 berikut.



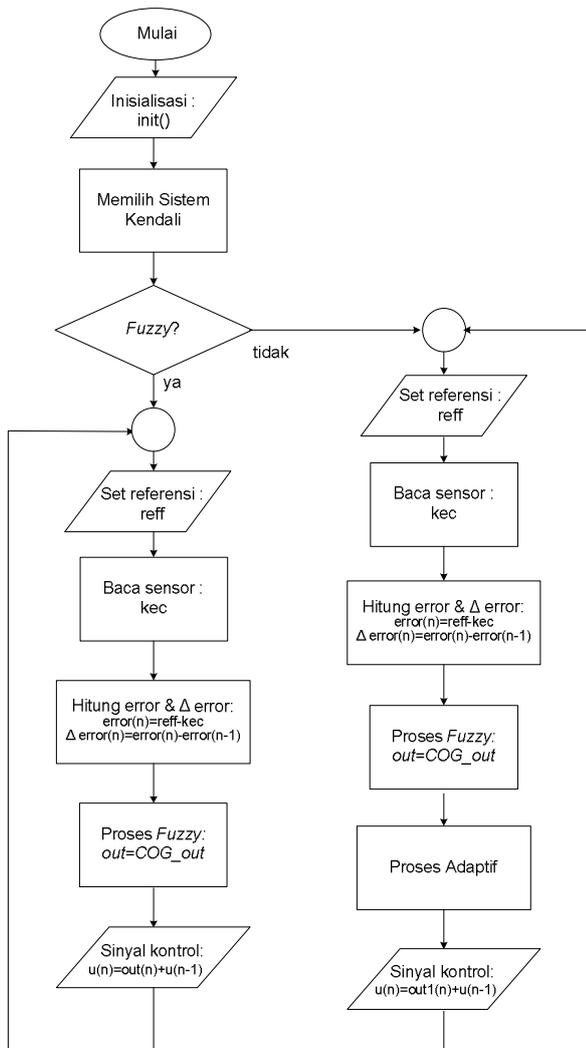
Gambar 3.5 Blok Diagram Sistem Kendali Fuzzy



Gambar 3.6 Blok Diagram Sistem Kendali Fuzzy Adaptif

### 3.2.1 Program Utama

Perancangan perangkat lunak pada perangkat ini meliputi pengaturan kerja sistem seperti pembacaan masukan (keypad), mengatur tampilan, pembacaan hasil sensor, pengaturan PWM, serta aksi-aksi lain yang dilakukan oleh perangkat. Gambar 3.10 berikut adalah gambar diagram alir pengaturan kecepatan motor DC pada program utama.



Gambar 3.7 Diagram Alir Program Utama

### 3.2.2 Sub Rutin Update dan Pembacaan Sensor Optocoupler

Sub rutin ini berfungsi untuk mengetahui kecepatan motor setiap saat terhadap referensinya. Besarnya waktu cuplik diatur lewat interupsi timer 0. Inisialisasi timer 0 sebagai berikut.

Besarnya waktu sampling dapat diubah-ubah dengan mengganti nilai batas variabel “waktu”. Waktu cuplik ini juga digunakan untuk membaca sensor dan update sinyal kendali. Dari subrutin di atas, maka besarnya waktu cuplik  $t$  dihitung sebagai berikut.

$$t = n_{\text{perulangan}} * \frac{(\text{nilai max 8bit} - \text{initawal})}{\text{clock}}$$

$$= 500 * \frac{(256 - 6)}{500k} = 0,25 \text{ detik}$$

Dan kecepatan motor yang dideteksi sensor optocoupler dihitung sebagai berikut.

$$\text{Kec}(\text{rpm}) = (\text{BIThighT} * 256 + \text{BITlowT} * 1) * \frac{60s}{t\text{Update} * \text{jmlLubang}}$$

$$= (\text{BIThighT} * 256 + \text{BITlowT} * 1) * \frac{60s}{0,25s * 40}$$

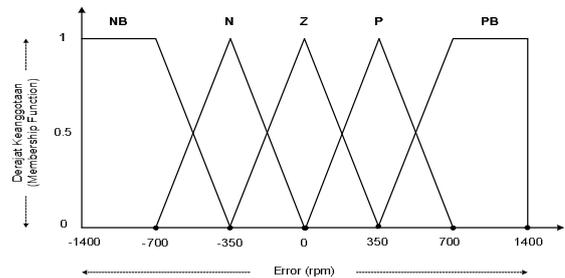
$$= (\text{BIThighT} * 256 + \text{BITlowT} * 1) * 6$$

### 3.2.3 Sub Rutin Kendali Logika Fuzzy

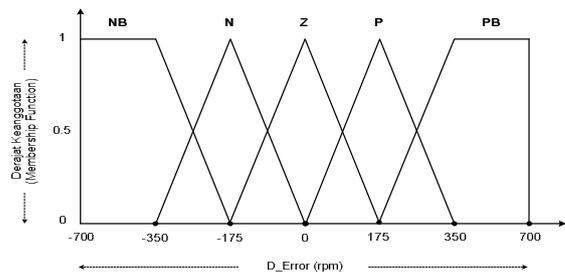
Pengendali logika fuzzy akan mengevaluasi setiap *input crisp* berupa *error* dan perubahan *error* ke dalam tiga tahapan, yaitu fuzzifikasi, evaluasi aturan dan defuzzifikasi.

#### A. Fuzzifikasi

Fuzzifikasi adalah proses pemetaan *input crisp* ke dalam himpunan-himpunan fuzzy dalam bentuk fungsi keanggotaan.



Gambar 3.8 Fungsi keanggotaan *error* jarak



Gambar 3.9 Fungsi keanggotaan *d\_error* jarak

#### B. Evaluasi aturan

Evaluasi aturan adalah proses mengevaluasi derajat keanggotaan tiap-tiap fungsi keanggotaan himpunan fuzzy masukan ke dalam basis aturan yang telah ditetapkan. Tujuan dari evaluasi aturan ini adalah menentukan derajat keanggotaan dari keluaran fuzzy.

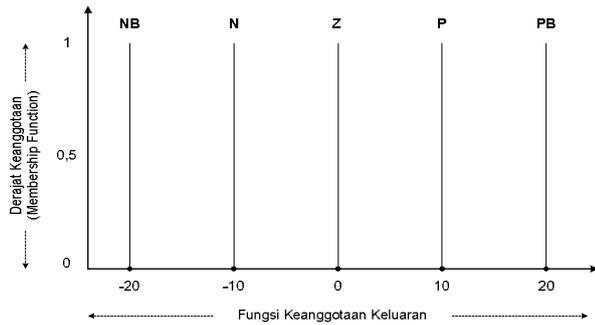
Tabel 3.1 Basis Aturan Kendali Logika Fuzzy

D_Error \ Error	NB	N	Z	P	PB
NB	NB	NB	NB	N	Z
N	NB	NB	N	Z	P
Z	NB	N	Z	P	PB
P	N	Z	P	PB	PB
PB	Z	P	PB	PB	PB

Metode pengambilan keputusan (inferensi) yang digunakan dalam pemrograman ini adalah metode Max-Min. Setelah semua aturan *fuzzy* dieksekusi, dilakukan proses agregasi dengan mengambil nilai maksimal dari masing-masing fungsi keanggotaan variabel keluaran.

### C. Defuzzifikasi

Tahap terakhir dari inferensi *fuzzy* adalah defuzzifikasi. Defuzzifikasi merupakan kebalikan dari proses fuzzifikasi, yaitu mengubah himpunan *fuzzy* keluaran menjadi keluaran tegas (*crisp*).



Gambar 3.10 Fungsi keanggotaan keluaran

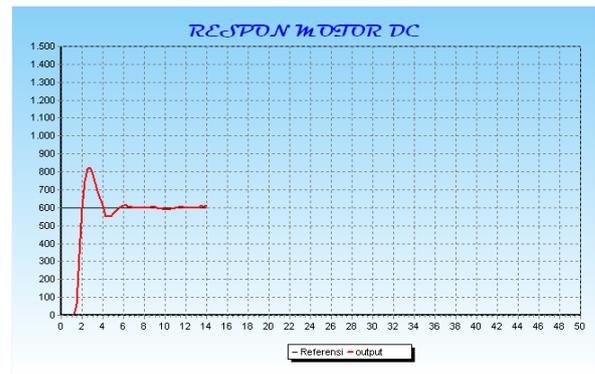
Untuk mendapatkan nilai tegas (*crisp*) keluaran dari himpunan-himpunan *fuzzy* keluaran, digunakan metode *Center of Gravity* (COG). Untuk pengaturan dengan kendali logika *fuzzy* adaptif metode tuning output digunakan blok adaptif yang akan mentuning keluaran dari kendali logika *fuzzy*.

## IV. PENGUJIAN DAN ANALISA

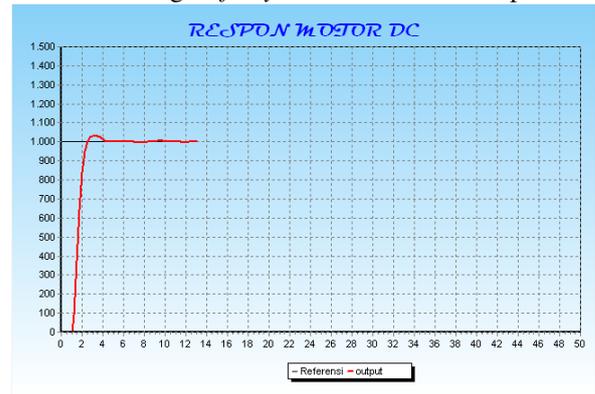
Pada tahap ini, perangkat diuji dengan variasi himpunan *fuzzy* keluaran dan variasi blok adaptif dari sistem kendali logika *fuzzy* adaptif, lalu diamati respon transient motor. Lewat pengujian ini hasilnya akan dianalisa untuk menentukan sistem kendali yang terbaik.

### 4.1 Pengujian Perangkat Menggunakan Sistem Kendali Logika *Fuzzy*

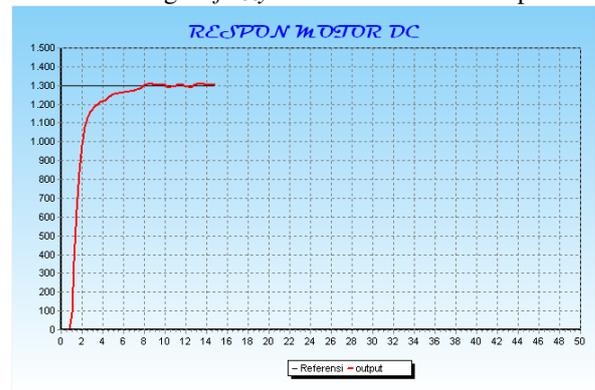
Untuk mengetahui respon transient dari motor, maka dilakukan percobaan dengan referensi kecepatan 600 rpm, 1000 rpm dan 1300 rpm. Respon transient untuk masing – masing referensi diamati dengan menggunakan program DELPHI 6 dapat dilihat pada gambar di bawah ini :



Gambar 4.1 Respon motor dengan sistem kendali logika *fuzzy* saat referensi = 600 rpm



Gambar 4.2 Respon motor dengan sistem kendali logika *fuzzy* saat referensi = 1000 rpm



Gambar 4.3 Respon motor dengan sistem kendali logika *fuzzy* saat referensi = 1300 rpm

Dari data pengujian diatas dapat diamati respon transient motor untuk setiap referensi. Data parameter respon transient dari hasil pengamatan adalah sebagai berikut.

Tabel 4.1 Parameter respon transient motor dengan sistem kendali logika *fuzzy*

Referensi (rpm)	Parameter					
	Td (s)	Tr (s)	Tp (s)	Ts (s)	Mp (rpm)	Ess (rpm)
600	1,75	2	2,5	7	222	0
1000	1,5	2,5	3	4,5	32	0
1300	2	3	-	8	-	0

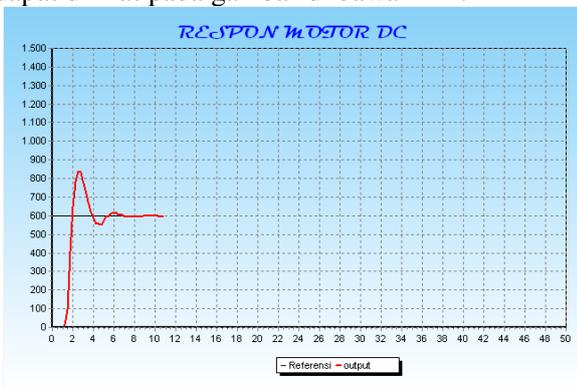
## 4.2 Perangkat Menggunakan Sistem Kendali Logika Fuzzy Adaptif

### 4.2.1 Percobaan Pertama

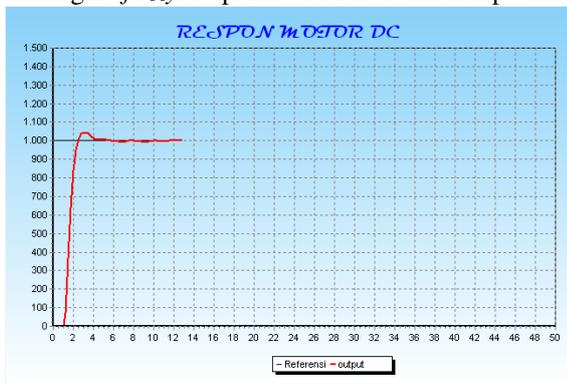
Aksi tuning output yang dilakukan blok adaptif dapat dilihat pada Tabel 4.2 berikut. Tabel 4.2 Tabel aksi tuning terhadap output Fuzzy Adaptif percobaan pertama

Error	Aksi tuning terhadap output
Kecil	Tetap
Sedang	Tetap
Besar	Memperbesar

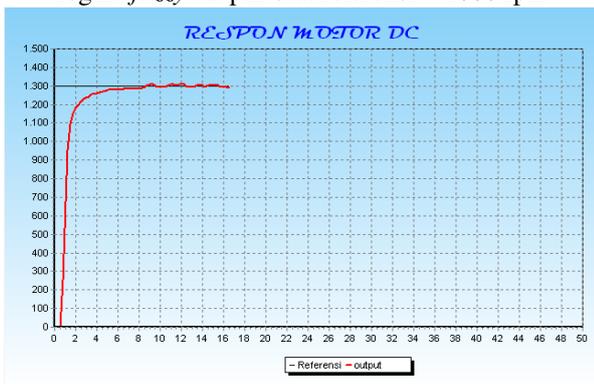
Untuk mengetahui respon transient dari motor, maka dilakukan percobaan dengan referensi kecepatan 600 rpm, 1000 rpm dan 1300 rpm. Respon transient untuk masing – masing referensi diamati dengan menggunakan program DELPHI 6 dapat dilihat pada gambar di bawah ini :



Gambar 4.4 Respon motor dengan sistem kendali logika fuzzy adaptif saat referensi = 600 rpm



Gambar 4.5 Respon motor dengan sistem kendali logika fuzzy adaptif saat referensi = 1000 rpm



Gambar 4.6 Respon motor dengan sistem kendali logika fuzzy adaptif saat referensi = 1300 rpm

Dari data pengujian diatas dapat diamati respon transient motor untuk setiap referensi. Data parameter respon transient dari hasil pengamatan adalah sebagai berikut.

Tabel 4.3 Parameter respon transient motor dengan sistem kendali logika fuzzy adaptif

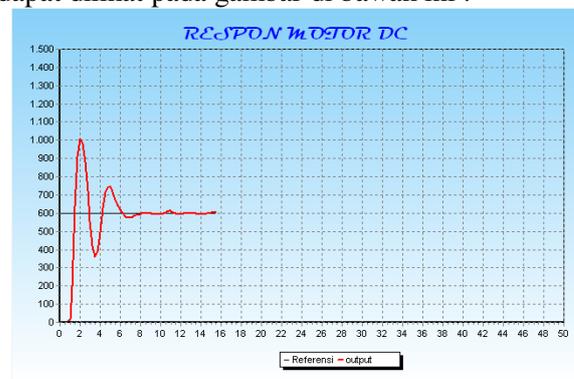
Referensi (rpm)	Parameter					
	Td (s)	Tr (s)	Tp (s)	Ts (s)	Mp (rpm)	Ess (rpm)
600	1,5	2	2,5	7	240	0
1000	1,5	2,5	3	4	44	0
1300	1	2	-	7	-	0

### 4.2.2 Percobaan Kedua

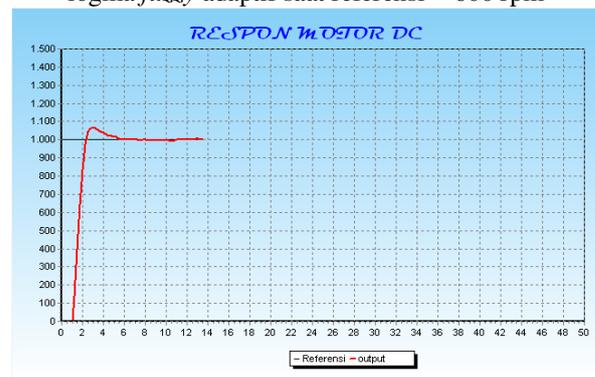
Aksi tuning output yang dilakukan blok adaptif dapat dilihat pada Tabel 4.4 berikut. Tabel 4.4 Tabel aksi tuning terhadap output Fuzzy Adaptif percobaan kedua

Error	Aksi tuning terhadap output
Kecil	Tetap
Sedang	Memperbesar
Besar	Tetap

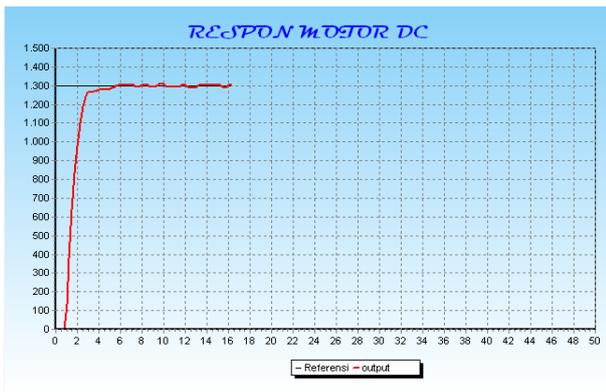
Untuk mengetahui respon transient dari motor, maka dilakukan percobaan dengan referensi kecepatan 600 rpm, 1000 rpm dan 1300 rpm. Respon transient untuk masing – masing referensi diamati dengan menggunakan program DELPHI 6 dapat dilihat pada gambar di bawah ini :



Gambar 4.7 Respon motor dengan sistem kendali logika fuzzy adaptif saat referensi = 600 rpm



Gambar 4.8 Respon motor dengan sistem kendali logika fuzzy adaptif saat referensi = 1000 rpm



Gambar 4.9 Respon motor dengan sistem kendali logika *fuzzy* adaptif saat referensi = 1300 rpm

Dari data pengujian diatas dapat diamati respon transient motor untuk setiap referensi. Data parameter respon transient dari hasil pengamatan adalah sebagai berikut.

Tabel 4.5 Parameter respon transient motor dengan sistem kendali logika *fuzzy* adaptif

Referensi (rpm)	Parameter					
	Td (s)	Tr (s)	Tp (s)	Ts (s)	Mp (rpm)	Ess (rpm)
600	1,25	1,5	2	8	408	0
1000	1,5	2,5	3	5,5	62	0
1300	1,5	2,5	-	6	-	0

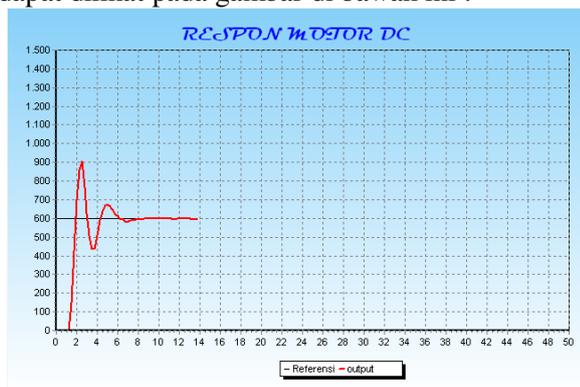
### 4.2.3 Percobaan Ketiga

Aksi tuning output yang dilakukan blok adaptif dapat dilihat pada Tabel 4.6 berikut.

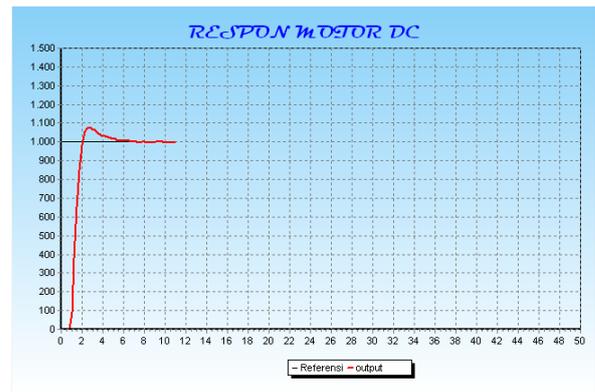
Tabel 4.6 Tabel aksi tuning terhadap output *Fuzzy* Adaptif percobaan ketiga

Error	Aksi tuning terhadap output
Kecil	Tetap
Sedang	Memperbesar
Besar	Memperbesar

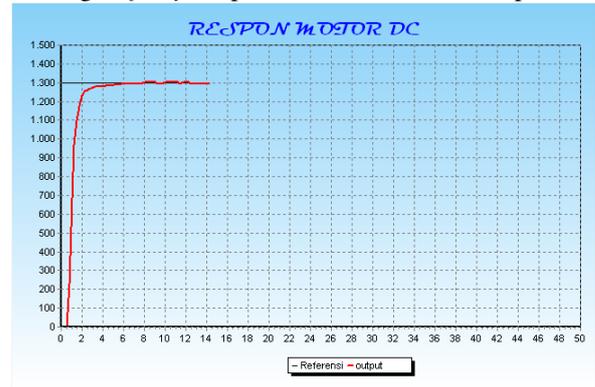
Untuk mengetahui respon transient dari motor, maka dilakukan percobaan dengan referensi kecepatan 600 rpm, 1000 rpm dan 1300 rpm. Respon transient untuk masing – masing referensi diamati dengan menggunakan program DELPHI 6 dapat dilihat pada gambar di bawah ini :



Gambar 4.10 Respon motor dengan sistem kendali logika *fuzzy* adaptif saat referensi = 600 rpm



Gambar 4.11 Respon motor dengan sistem kendali logika *fuzzy* adaptif saat referensi = 1000 rpm



Gambar 4.12 Respon motor dengan sistem kendali logika *fuzzy* adaptif saat referensi = 1300 rpm

Dari data pengujian diatas dapat diamati respon transient motor untuk setiap referensi. Data parameter respon transient dari hasil pengamatan adalah sebagai berikut.

Tabel 4.7 Parameter respon transient motor dengan sistem kendali logika *fuzzy* adaptif

Referensi (rpm)	Parameter					
	Td (s)	Tr (s)	Tp (s)	Ts (s)	Mp (rpm)	Ess (rpm)
600	1,5	2	2,5	7,5	306	0
1000	1,5	2	2,5	6	80	0
1300	1	2	-	6	-	0

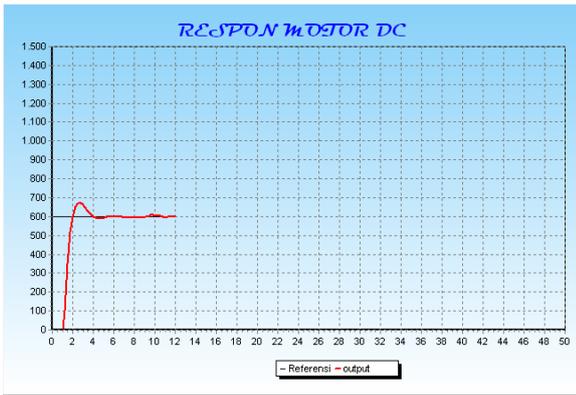
### 4.2.4 Percobaan Keempat

Aksi tuning output yang dilakukan blok adaptif dapat dilihat pada Tabel 4.8 berikut.

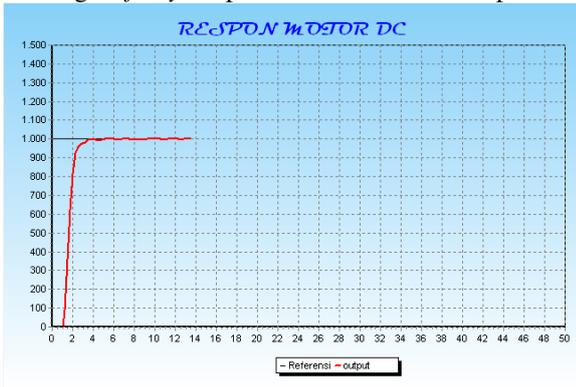
Tabel 4.8 Tabel aksi tuning terhadap output *Fuzzy* Adaptif percobaan keempat

Error	Aksi tuning terhadap output
Kecil	Tetap
Sedang	Memperkecil
Besar	Tetap

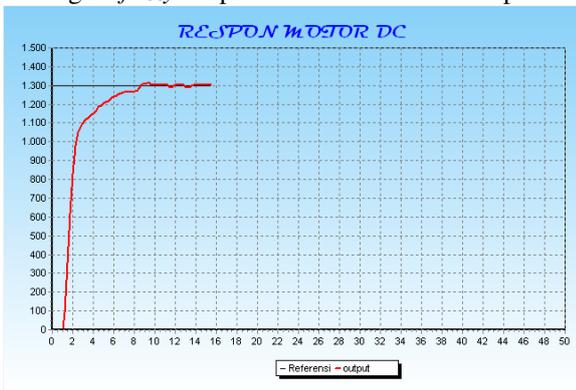
Untuk mengetahui respon transient dari motor, maka dilakukan percobaan dengan referensi kecepatan 600 rpm, 1000 rpm dan 1300 rpm. Respon transient untuk masing – masing referensi diamati dengan menggunakan program DELPHI 6 dapat dilihat pada gambar di bawah ini :



Gambar 4.13 Respon motor dengan sistem kendali logika *fuzzy* adaptif saat referensi = 600 rpm



Gambar 4.14 Respon motor dengan sistem kendali logika *fuzzy* adaptif saat referensi = 1000 rpm



Gambar 4.15 Respon motor dengan sistem kendali logika *fuzzy* adaptif saat referensi = 1300 rpm

Dari data pengujian diatas dapat diamati respon transient motor untuk setiap referensi. Data parameter respon transient dari hasil pengamatan adalah sebagai berikut.

Tabel 4.9 Parameter respon transient motor dengan sistem kendali logika *fuzzy* adaptif

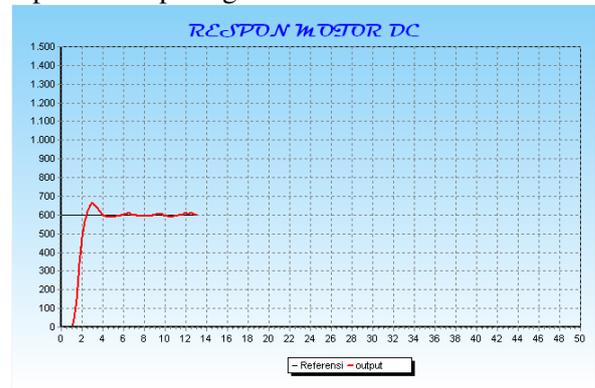
Referensi (rpm)	Parameter					
	Td (s)	Tr (s)	Tp (s)	Ts (s)	Mp (rpm)	Ess (rpm)
600	1,5	2	2,5	4,5	72	0
1000	1,5	2,25	-	4	-	0
1300	1,75	4	-	9	-	0

#### 4.2.5 Percobaan Kelima

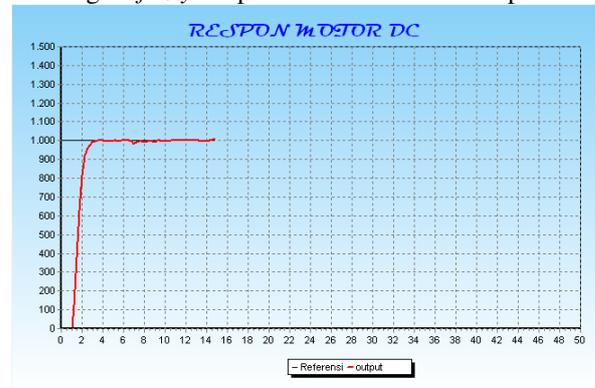
Aksi tuning output yang dilakukan blok adaptif dapat dilihat pada Tabel 4.10 berikut. Tabel 4.10 Tabel aksi tuning terhadap output *Fuzzy* Adaptif percobaan kelima

Error	Aksi tuning terhadap output
Kecil	Tetap
Sedang	Memperkecil
Besar	Memperbesar

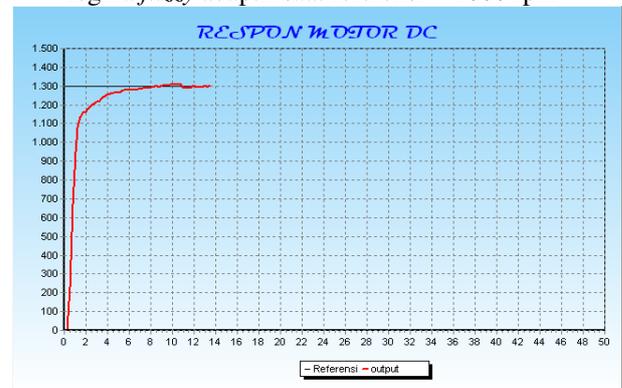
Untuk mengetahui respon transient dari motor, maka dilakukan percobaan dengan referensi kecepatan 600 rpm, 1000 rpm dan 1300 rpm. Respon transient untuk masing – masing referensi diamati dengan menggunakan program DELPHI 6 dapat dilihat pada gambar di bawah ini :



Gambar 4.16 Respon motor dengan sistem kendali logika *fuzzy* adaptif saat referensi = 600 rpm



Gambar 4.17 Respon motor dengan sistem kendali logika *fuzzy* adaptif saat referensi = 1000 rpm



Gambar 4.18 Respon motor dengan sistem kendali logika *fuzzy* adaptif saat referensi = 1300 rpm

Dari data pengujian diatas dapat diamati respon transient motor untuk setiap referensi. Data parameter respon transient dari hasil pengamatan adalah sebagai berikut.

Tabel 4.11 Parameter respon transient motor dengan sistem kendali logika *fuzzy* adaptif

Referensi (rpm)	Parameter					
	Td (s)	Tr (s)	Tp (s)	Ts (s)	Mp (rpm)	Ess (rpm)
600	1,5	2,25	2,75	4	66	0
1000	1,5	2,25	-	3	-	0
1300	1	2	-	7	-	0

Pengujian menggunakan kendali logika *fuzzy* adaptif metode tuning output dilakukan melalui lima kali percobaan. Dari lima kali percobaan diambil secara empiris, aksi tuning output yang paling tepat.

## V. PENUTUP

### 5.1 Kesimpulan

Berdasarkan perancangan, pengujian, dan analisis yang telah dilakukan, maka dapat disimpulkan hal-hal sebagai berikut:

1. Sistem kendali logika *fuzzy* adaptif pada percobaan pertama belum memperbaiki respon motor daripada menggunakan sistem kendali logika *fuzzy*. Proses adaptif pada sistem kendali ini memperbesar keluaran kendali logika *fuzzy* saat errornya besar sehingga memperbesar *overshoot* yang terjadi.
2. Sistem kendali logika *fuzzy* adaptif pada percobaan kedua belum memperbaiki respon motor daripada menggunakan sistem kendali logika *fuzzy*. Proses adaptif pada sistem kendali ini memperbesar keluaran kendali logika *fuzzy* saat errornya sedang sehingga memperbesar *overshoot* dan menimbulkan osilasi saat nilai referensinya kecil.
3. Sistem kendali logika *fuzzy* adaptif pada percobaan ketiga belum memperbaiki respon motor daripada menggunakan sistem kendali logika *fuzzy*. Proses adaptif pada sistem kendali ini memperbesar keluaran kendali logika *fuzzy* saat errornya sedang atau besar sehingga memperbesar *overshoot* dan menimbulkan osilasi.
4. Sistem kendali logika *fuzzy* adaptif pada percobaan keempat belum memperbaiki respon motor daripada menggunakan sistem kendali logika *fuzzy*. Proses adaptif pada

sistem kendali ini memperkecil keluaran kendali logika *fuzzy* saat errornya sedang sehingga membuat respon motor menjadi lamban.

5. Sistem kendali logika *fuzzy* adaptif pada percobaan kelima dapat memperbaiki respon motor daripada menggunakan sistem kendali logika *fuzzy*. Proses adaptif pada sistem kendali ini memperbesar keluaran kendali logika *fuzzy* saat errornya besar sehingga mempercepat respon sistem dan memperkecil keluaran kendali logika *fuzzy* saat errornya sedang sehingga menekan serta menghilangkan *overshoot* dan osilasi.
6. Dengan membandingkan kedua sistem kendali didapatkan bahwa kendali logika *fuzzy* adaptif metode tuning output menghasilkan respon sistem yang lebih baik daripada kendali logika *fuzzy*.

### 5.2 Saran

Sebagai masukan guna pengembangan lebih lanjut dari Tugas Akhir ini, maka penulis memberikan beberapa saran sebagai berikut :

1. Dapat ditambahkan himpunan *fuzzy* untuk masukan dan keluaran untuk memperbaiki kinerja sistem kendali logika *fuzzy* adaptif.
2. Sistem kendali logika *fuzzy* adaptif dapat dikembangkan dengan mengaplikasikannya, misalnya untuk sistem conveyor atau robot *mobile*.

## DAFTAR PUSTAKA

- [1] Kristanto, Andri, *Struktur Data Dengan C++*, Graha Ilmu, 2003.
- [2] Jyh Shing Roger Jang, Chuen Tsai Sun, Eiji Mizutani. *Neuro Fuzzy and Soft Computing*, Prentice-Hall International, Inc, 1997.
- [3] Ogata, Katsuhiko, *Teknik Kontrol Automatik Jilid 1-2*, Diterjemahkan Oleh Ir. Edi Leksono, Penerbit Erlangga, Jakarta, 1996.
- [4] Hartono, Jogiyanto, *Konsep Dasar Pemrograman Bahasa C*, Penerbit Andi, Yogyakarta, 1992.
- [5] Benjamin C. Kuo, *Automatic Control System Sixth Edition*, Prentice-Hall, New Jersey, 1991.
- [6] Jun Yan, Michael Ryan, James Power, *Using Fuzzy Logic*, Prentice-Hall, Cambridge, 1994.
- [7] Shigley, Joseph Edward, Charles R. Mischke, *Mechanical Engineering Design Fifth Edition*, McGraw-Hill, Singapore, 1989.

- [8] Wang Li-Xin, *A Course in Fuzzy Systems and Control*. Prentice-Hall International Inc, New Jersey, 1997.
- [9] Muslimin M, *Teknik Tenaga Listrik (TTL)*, Armico, Bandung, 1979.
- [10] -----, ATmega8535 Data Sheet, <http://www.atmel.com>.
- [11] -----, L293D Data Sheet, <http://www.ti.com>.

**Peter Christianto (L2F 003 526)**



Lahir di Semarang, 10 November 1985. Saat ini sedang melanjutkan studi pendidikan strata I di Jurusan Teknik Elektro, Fakultas Teknik Universitas Diponegoro Konsentrasi Kontrol.

Mengetahui dan mengesahkan,

Dosen Pembimbing I

Dosen Pembimbing II

Darjat, ST, MT  
NIP. 132 231 135

Iwan Setiawan, ST, MT  
NIP. 132 283 183

Tanggal: \_\_\_\_\_

Tanggal: \_\_\_\_\_