# PENGATURAN POSISI MOTOR SERVO DC DENGAN METODE FUZZY LOGIC

Rois' Am[1], Kemalasari[2], Bambang Sumantri[2], Ardik Wijayanto[2]
[1]Penulis,Mahasiswa Jurusan Teknik Elektronika PENS-ITS
[2]Dosen Pembimbing, Staf Pengajar di Jurusan Teknik Elektronika PENS-ITS
Politeknik Elektronika Negeri Surabaya
Electronics Engineering Polytechnic Institute of Surabaya
Institut Teknologi Sepuluh November
Kampus ITS Sukalilo, Surabaya 60111,INDONESIA
Tel: +62(31) 594 7280; Fax: +62(31) 594 6114
email: masiroisam@gmail.com

***Abstrak***

Dari beberapa jenis motor, motor servo adalah salah satu jenis motor yang memiliki keunikan. Keunikan tersebut adalah motor servo memiliki torsi yang besar dan putaran yang lambat. Oleh karena itu motor servo banyak digunakan dibidang robotika. Sedangkan pada jurusan elektronika adalah jurusan yang mengutamakan bidang robotika dan sistem kontrol. Sehingga dengan adanya proyek akhir ini dapat membantu memudahkan mahasiswa dalam memberikan pemahaman tentang pengaturan posisi pada motor servo.

Pengaturan posisi disini menggunakan metode logika fuzzy. Digunakannya metode fuzzy bertujuan untuk memperbaiki sistem kontrol klasik yang cenderung rumit jika dibandingkan dengan fuzzy. Pada proyek akhir ini dilakukan pengujian dengan memberikan masukan sudut melalui PC dengan PCI 1712 sebagai penghubung antara motor dengan PC. Masukan yang diperbolehkan adalah semua sudut yang berada antara 0 sampai 180. Sistem yang dihasilkan memiliki *time seting* 0,5 hingga 3,2 sekon dan *maximum overshoot* 0 hingga $2,5^0$. Dengan adanya beban, *time seting* dapat mencapai 4,5 sekon.

**Kata kunci :** motor servo, *fuzzy logic*, PCI 1712

## I. PENDAHULUAN

Akhir-akhir ini motor servo sangat banyak digunakan dalam bidang robotika. Motor servo ini biasa digunakan sebagai sistem gerak dalam *fixed robot* dan *mobile robot.* Oleh karena itu diperlukan pemahaman yang mendalam tentang motor servo, termasuk pemahaman tentang pergerakan atau perubahaan posisi dari motor servo. Sedangkan pengaturan posisi motor servo ini dibutuhkan kontroler untuk meminimalkan presentase kesalahan dari posisi yang dikehendaki dengan posisi yang dihasilkan.

Dari pentingnya pemahaman tentang motor servo ini sehingga kami akhirnya mengangkat topik ini sebagai judul dari proyek akhir kami. Pada proyek akhir ini kami akan membuat permodelan sistem motor servo yang dibentuk dari motor DC standart yang dihubungkan dengan sistem gaer. Agar setiap perpindahan sudut dari motor servo dapat diamati maka output motor dihubungkan dengan sebuah busur derajat.

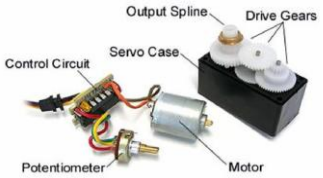
Pada proyek akhir ini kami akan menggunakan metode *Fuzzy Logic* untuk menggatur posisi sesuai dengan kehendak pengguna. Digunakannya metode fuzzy untuk memudahkan penggontrolan jika dibandingkan dengan kontroler klasik seperti kontroler PID.

## II. DASAR TEORI

### 2.1 MOTOR DC SERVO

Motor servo adalah sebuah motor dengan sistem umpan balik tertutup di mana posisi dari motor akan diinformasikan kembali ke rangkaian kontrol yang ada di dalam motor servo. Motor ini terdiri dari sebuah motor DC, serangkaian gear, potensiometer dan rangkaian kontrol. Potensiometer berfungsi untuk menentukan batas sudut dari putaran servo. Sedangkan sudut dari sumbu motor servo diatur berdasarkan lebar pulsa yang dikirim melalui kaki sinyal dari kabel motor[1].

Seperti yang kita tahu bahwa servo terdiri dari rangkaian pengontrol, gear, potensiometer dan DC motor. Potensiometer terhubung dengan gear demikian pula DC motor. Ketika DC motor diberi signal oleh rangkaian pengontrol maka dia akan bergerak demikian pula potensiometer dan otomatis akan mengubah resistansinya. Rangkaian pengontrol akan mengamati perubahan resistansi dan ketika resistansi mencapai nilai yang diinginkan maka motor akan berhenti pada posisi yang diinginkan.




**Gambar 2.1** Bentuk Motor Servo [2]

### 2.2 FUZZY LOGIC

Kontroller *fuzzy logic* dikategorikan dalam kontrol cerdas (*intelligent control*). Dimana Fuzzy berarti kabur atau samar (kualitaif) dan Logic berarti "umumnya dilakukan orang yaitu berpikir secara logis". Jadi, *Fuzzy logic* berarti berpikir secara logika untuk parameter yang kualitatif (samar).


**Gambar 2.2** Model "*Black Box*"

Gambar 2.2 memberikan ilustrasi pemetaan hubungan *input-output*. Diantara *input* dan *output* kita taruh sebuah sistem *black box* yang akan melakukan tugas pemetaan. Sistem yang cocok menggantikan kotak hitam tersebut ada banyak alternatif seperti sistem fuzzy, linier, sistem jaringan saraf tiruan dan masih banyak lagi. [3]

Kontroller yang berbasis *fuzzy logic* harus melalui beberapa tahapan sebelum sampai ke plant. Tahapa tersebut antara lain : Kuantisasi, Fuzzifikasi, Penentuan Rule, kemudian Defuzzifikasi.

- **Kuantisasi**

Proses pengambilan masukan suatu numerik *input* missal masukan error dan delta error, kemudian mengubah menjadi tingkat kuantisasi. Jumlah tingkat kuantisasi akan menentukan ketelitian dalam pengambilan keputusan, sehingga makin banyak tingkat kuantisasi hasil ketelitian akan lebih baik, tetapi perhitungan akan semakin rumit. Untuk sistem yang sederhana cukup menggunakan sedikit jumlah tingkat kuantisasi. Klasifikasi ditentukan dalam bentuk Negatif, Nol, dan Positif, maka tingkat kuantisasi yaitu (-1,0,1). Hal ini dapat diperluas lagi dengan menambah tingkat kuantisasi (-2,-1,0,1,2), maka klasifikasi terdiri dari Negatif Besar (NB), Negatif kecil (Nk), Nol (Z), Positif kecil (Pk), Positif Besar (PB).

- **Fuzzifikasi**

Prosedur fuzzifikasi merupakan proses untuk mengubah variabel non fuzzy (variabel numerik) menjadi variabel fuzzy (variable linguistik). Nilai error dan delta error yang dikuantisasi sebelumnya diolah oleh kontroler *fuzzy logic*, kemudian diubah terlebih dahulu ke dalam variabel fuzzy. Melalui membership function (fungsi keanggotaan) yang telah disusun, maka dari nilai error dan delta error kuantisasi akan didapatkan derajat keanggotaan bagi masing-masing nilai error dan delta error. Pada unit fuzzifikasi ini terjadi proses transformasi, yang dilakukan dengan cara pemetaan ruang masukan, dari variabel masukan domain non-fuzzy (*crisp*) ke dalam domain fuzzy, dengan bantuan faktor penskala (*scaling factor*). Faktor penskala menggunakan metode heuristik, diatur sedemikian rupa sehingga seluruh variabel masukan terpetakan dalam semesta pembicaraan yang dirancang. Perbedaan kecepatan (error) diskala dengan gain G1, sedangkan perbedaan perubahan kecepatan (delta error) diskala dengan gain G2.

- **Rule**

Basis pengetahuan fuzzy terdiri dari beberapa aturan fuzzy yang dikelompokkan kedalam suatu basis aturan, disebut basis aturan fuzzy (fuzzy rule base). Rule base merupakan dasar dari pengambilan keputusan atau *inference process*, untuk mendapatkan aksi keluaran sinyal kontrol dari suatu kondisi masukan yaitu error dan delta error, dengan berdasarkan rule-rule yang telah ditetapkan. Proses inferensi menghasilkan sinyal keluaran yang masih dalam bentuk bilangan fuzzy, yaitu derajat keanggotaan dari sinyal kontrol. Pendefinisian rule base tergantung dari data yang telah disesuaikan dan dalam bentuk tabel. Sebagai contoh; Rule base dengan 5 tingkat kuantisasi untuk variabel *input* (error E dan delta error dE)

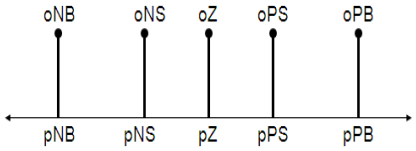
- **Defuzzifikasi**

Defuzzifikasi merupakan kebalikan dari proses transformasi sebuah himpunan fuzzy kedalam himpunan tegas. Metode defuzzifikasi yang umum digunakan ada 2 macam yaitu : [3]

- Metode Mamdani,center of gravity

$$y = \frac{\sum_{j=1}^{R} c^j \int \mu_{\tilde{D}J}(u)\, du}{\sum_{j=1}^{R} \int \mu_{\tilde{D}J}(u)\, du} \quad \text{.....(2.1)}$$

- Metode Sugeno, single tone



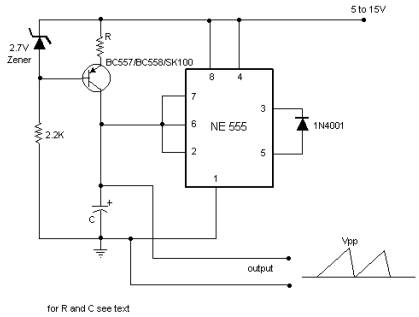

**Gambar 2.3** Out Single Tone

$$Out = \frac{oNB.pNB + oNS.pNS + oZ.pZ + oPS.pPS + oPB.pPB}{oNB + oNS + oZ + oPS + oPB}$$

$$Out = \frac{\sum_{i=1}^{N} O_i.p_i}{\sum_{i=1}^{N} O_i} \quad \text{.....(2.2)}$$

## 2.3 PEMBANGKIT GELOMBANG GIGI GERGAJI

Sebuah generator gelombang gigi gergaji dapat dibangun dengan menggunakan IC 555 timer sederhana dan sebuah transistor seperti ditunjukkan pada gambar dibawah ini.




**Gambar 2.4** Rangkaian pembangkit gelombang gigi gergaji

Perhatikan bahwa output diambil di kapasitor. Dioda 1N4001 membuat tegangan kapasitor pergi ke permukaan *ground*. Frekuensi sirkuit diperoleh dari:

$$f = \frac{(V_{cc} - V_{zener})}{R \; x \; C \; x \; V_{pp}} \quad \text{.....(2.3)}$$

dimana:

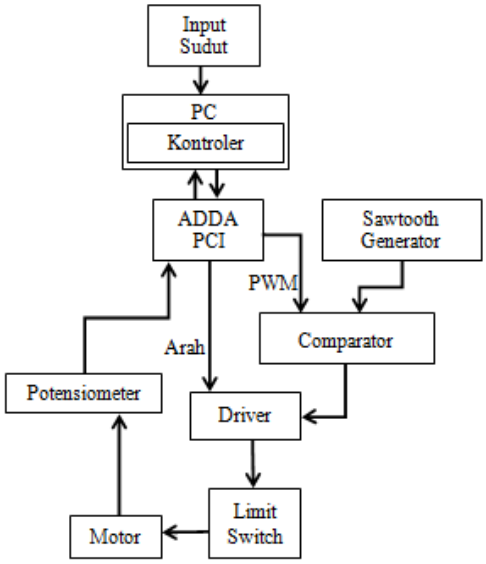
Vcc = Suplai tegangan.

Vpp = *Peak to peak* tegangan output yang dibutuhkan.[4]

## III. PERANCANGAN SISTEM

### 3.1 TAHAP PERANCANGAN

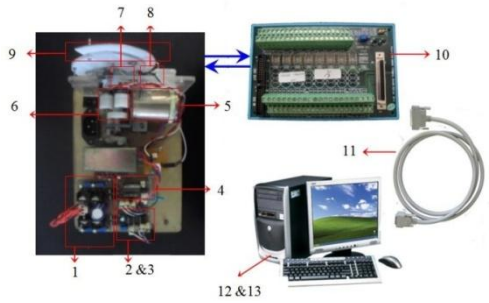
Dalam tahap ini akan dibahas bagaimana sistem tersebut beroperasi dan komponen-komponen yang menyusun system ini :




**Gambar 3.1** Blok diagram sistem

### 3.2 PEMBUATAN PERANGKAT KERAS

Perangkat keras (*hardwre*) merupakan syarat utama dari sistem yang akan dirancang. Komponen utama sistem yang akan dibuat pada proyek akhir ini akan dijelaskan lebih rinci. Berikut adalah gambaran peletakkan tiap blok dari sistem

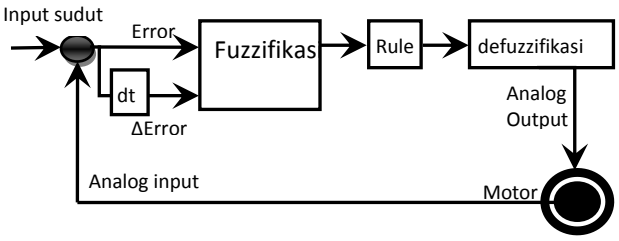



**Gambar 3.2** Gambaran sistem secara umum

Keterangan gambar :
1. Catu daya
2. *Sawtooth generator*
3. Komparator
4. *Driver* motor
5. Motor
6. *Gear box*
7. Potensiometer
8. *Limit switch*
9. Tampilan busur
10. PCLD8712
11. kabel PCI
12. PCI 1712 (AD/DA)
13. PC

### 3.3 PEMBUATAN PERANGKAT LUNAK

Program kontroler fuzzy dibuat untuk mengatur besarnya tegangan analog *output* dengan masukan berupa error dan Δerror. Berikut blok diagram proses fuzzynya.




**Gambar 3.3** blok diagram sistem fuzzy pengaturan posisi

*input* fuzzy berupa error dan Δerror. Nilai error dan Δerror diperoleh dari:

Error = *Input* sudut – Analog *input* (3.1)

ΔError = Error sekarang – Error sebelumnya (3.2)

a) Fuzzifikasi

Pada fuzzifikasi ini, *input* error dan Δerror masing- masing dibagi dalam 5 fungsi keanggotaan sebagai berikut:

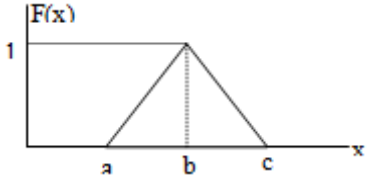
- Error: SK(sangat kecil), AK(agak kecil), S(sedang), AB(agak besar) dan SB(sangat besar)
- ΔError: NB(negatif besar), NK(negatif kecil), Z(zero), PK(positif kecil) dan PB(positif besar)

Sedangkan tipe keanggotaan fuzzy yang digunakan dalam proyek akhir ini ada 2 yaitu segitiga dan trapesium. Berikut ini persamaan dari segitiga dan trapesium:

Tipe keanggotaan segitiga:

$$F(x;a,b,c)=\begin{cases}0 & x<a\\ (x-a)/(b-a) & a\le x<b\\ (c-x)/(c-d) & b\le x<c\\ 0 & x\ge c\end{cases}$$

Dengan nilai x, a, b dan c diperoleh dari gambar berikut:




**Gambar 3.11** tipe keanggotaan segitiga

Nilai x ini merupakan nilai error atau Δerror yang dimasukkan dan F(x) adalah nilai keanggotaan. Dalam proyak akhir ini, *membership* yang bertipekan segitiga yaitu:

NK (x; -0,25; -0,125; 0) → Negatif Kecil

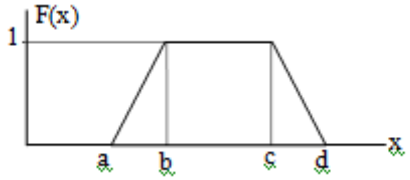
Z (x; -0,125; 0; 0,125) → Zero

PK (x; 0; 0,125; 0,25) → Positif Kecil

Tipe keanggotaan trapesium:

$$F(x;a,b,c,d)=\begin{cases}0 & x<a\\ (x-a)/(b-a) & a\le x<b\\ 1 & b\le x<c\\ (c-x)/(c-d) & c\le x<d\\ 0 & x\ge d\end{cases}$$

Dengan nilai x, a, b, c dan d diperoleh dari gambar berikut:

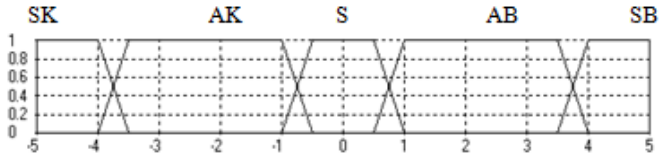



**Gambar 3.12** tipe keanggotaan trapesium

Nilai x ini merupakan nilai error atau Δerror yang dimasukkan dan F(x) adalah nilai keanggotaan. Dalam proyak akhir ini, membership yang bertipekan trapesium yaitu:

| | | |
|---|---|---|
| SK | (x; -5; -5; -4; -3,5) | → Sangat Kecil |
| AK | (x; -4; -3,5; -1; -0,5) | → Agak Kecil |
| S | (x; -1; -0,5; 0,5; 1) | → Sedang |
| AB | (x; 0,5; 1; 3,5; 4) | → Agak Besar |
| SB | (x; 3,5; 4; 5; 5) | → Sangat Besar |
| NB | (x; -5; -5; -1; -0,125) | → Negatif Besar |
| PB | (x; 0,125; 1; 5; 5) | → Positif Besar |

Berikut adalah perencanaan untuk keanggotaan error

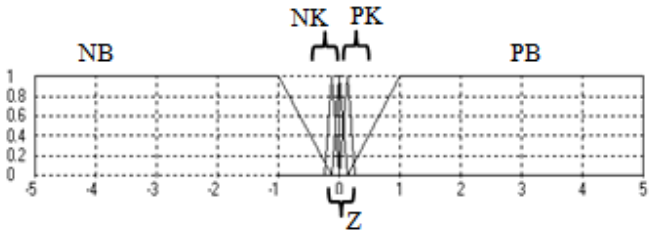



**Gambar 3.13** Perancangan *membership function* error

Dari rancangan diatas, dapat disusun program implementasi dalam visual basic sebagai berikut :

```
'Fuzzifikasi error
   ERRSK = TRAP(Val(TxErr), -5, -5, -4, -3.5)
   ERRK = TRAP(Val(TxErr), -4, -3.5, -1, -0.5)
   ERRS = TRAP(Val(TxErr), -1, -0.5, 0.5, 1)
   ERRB = TRAP(Val(TxErr), 0.5, 1, 3.5, 4)
   ERRSB = TRAP(Val(TxErr), 3.5, 4, 5, 5)
Function TRAP(u, a, B, C, d As Single) As Single
   If ((u > a) And (u < B)) Then
      TRAP = ((u - a) / (B - a))
   ElseIf ((u >= B) And (u <= C)) Then
      TRAP = 1
   ElseIf ((u > C) And (u < d)) Then
      TRAP = ((d - u) / (d - C))
   Else: TRAP = 0
   End If
```

Berikut adalah perencanaan untuk keanggotaan Δerror




**Gambar 3.14** Perancangan *membership function* Δerror

Dari rancangan diatas, dapat disusun program implementasi dalam visual basic sebagai berikut :

```
   DELNB = TRAP(Val(TxDerr), -5, -5, -1, -0.125)
   DELNK = Segitiga(Val(TxDerr), -0.25, -0.125, 0)
   DELZ = Segitiga(Val(TxDerr), -0.125, 0, 0.125)
   DELPK = Segitiga(Val(TxDerr), 0, 0.125, 0.25)
   DELPB = TRAP(Val(TxDerr), 0.125, 1, 5, 5)

Function Segitiga(u, a, B, C As Single) As Single
   If ((u >= a) And (u < B)) Then
      Segitiga = ((u - a) / (B - a))
   ElseIf ((u >= B) And (u < C)) Then
      Segitiga = ((C - u) / (C - B))
   Else: Segitiga = 0
   End If
End Function
Function TRAP(u, a, B, C, d As Single) As Single
   If ((u > a) And (u < B)) Then
      TRAP = ((u - a) / (B - a))
   ElseIf ((u >= B) And (u <= C)) Then
      TRAP = 1
   ElseIf ((u > C) And (u < d)) Then
      TRAP = ((d - u) / (d - C))
   Else: TRAP = 0
   End If
End Function
```

c) Rule Base

Dari membership function yang sudah dibuat, maka dapat dirumuskan rule fuzzifikasinya sesuai kondisi yang ada. Karena membership function *input*nya masing-masing memiliki lima buah member untuk masukan error dan Δerror. Sehingga didapatkan 25 rule kemungkinan yang dikombinasikan dengan 5 buah membership *output*. Berikut rule yang dirancang:

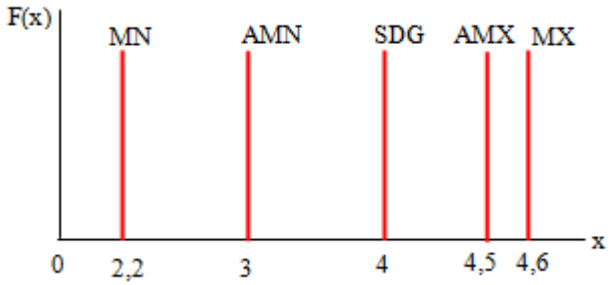
**Tabel 3.1** rule base hasil rancangan

| ERROR \ ΔERROR | NB | NK | Z | PK | PB |
|---|---|---|---|---|---|
| SK | MX | AMN | MN | AMN | MX |
| AK | AMX | AMN | MN | AMN | AMX |
| S | SDG | MN | MN | MN | SDG |
| AB | AMX | AMN | MN | AMN | AMX |
| SB | MX | AMN | MN | AMN | MX |

Dasar dari penentuan rule diatas adalah saat nilai error dan Δerror besar maka *output* yang dihasilkan maximal. Sedangkan saat error dan Δerror mendekati nol maka *output* minimal. Dari rancangan diatas, dapat disusun program implementasi dalam visual basic sebagai berikut :

```
R(1) = Minimal(ERRSK, DELNB) * MX
R(2) = Minimal(ERRSK, DELNK) * AMX
R(3) = Minimal(ERRSK, DELZ) * SDG
R(4) = Minimal(ERRSK, DELPK) * AMX
R(5) = Minimal(ERRSK, DELPB) * MX
R(6) = Minimal(ERRK, DELNB) * AMN
R(7) = Minimal(ERRK, DELNK) * AMN
R(8) = Minimal(ERRK, DELZ) * AMN
...
R(25) = Minimal(ERRSB, DELPB) * MX
```

Dimana nilai- nilai dari MN, AMN, SDG, AMX, MX seperti yang ditunjukkan grafik dibawah ini:




**Gambar 3.15** Nilai single tone *output* dari fuzzy

MN (Minimal) = 2,2
AMN (Agak Minimal) = 3
SDG (Sedang) = 4
AMX (Agak Maximal) = 4,5
MX (Maximal) = 4,6

b) Defuzzifikasi

Defuzzifikasi yang digunakan dalam proyak akhir ini menggunakan penalaran fuzzy metode sugeno. Digunakannya metode ini karena dalam sistem dibutuhkan proses yang cepat dan berlangsung kontinyu. Berikut ini rumusan dasar penalaran fuzzy metode sugeno.

$$Nilai\ Fuzzy = \sum_{i=1}^{n} \frac{\mu(k_n) x k_n}{\mu(k_n)} \qquad (3.4)$$

$\mu(k_n)$ adalah nilai dari proses AND antara *input* error dan *input* Δerror. Sedangkan $k_n$ adalah nilai *single tone output*

## 4. PENGUJIAN DAN ANALISA

### 4.1 Pengujian *membership, rule base dan output fuzzy logic* yang dirancang

**Tabel 4.1** nilai *seting time* dan maximum *overshoot* dari perubahan *rule*

| Rule ke | Perubahan | *Time seting* | *Overshoot* |
|---|---|---|---|
| - | - | 1 s | $2^0$ |
| 24 | AMX→MN | 2,8 s | $4^0$ |
| 24<br>13 | AMX→MN<br>MN→MX | ~ | $15^0$ |
| 14<br>19 | AMN→SDG<br>AMN→SDG | 1,8 s | $2^0$ |
| 13 | MN→AMN | 0,8 s | $2^0$ |

dari tabel diatas dapat diketahui bahwa penggunaan *rule base* dari perencanaan sudah tepat karena jika dibandingkan dengan lainnya maka nilai *time seting* dan *overshoot* memiliki nilai terkecil. Sehingga *rule base* dari perencanaan tidak perlu diubah.

**Tabel 4.2** nilai *seting time* dan maximum *overshoot* dari perubahan *membership function input*

| No | Perubahan | *Time seting* | *Overshoot* |
|---|---|---|---|
| 1 | - | 1 s | $2^0$ |
| 2 | AB (x; 0,5; 1; 1,1; 1.4)<br>SB (x; 1,1; 1,4; 5; 5) | 0,8 s | $10^0$ |
| 3 | S (x; -1; -0,5; 2,5; 3)<br>AB (x; 2,5; 3; 3,5; 4) | 4,8 s | $2^0$ |
| 4 | NB (x; -5; -5; -4; -3,5)<br>NK (x; -4; -2; 0)<br>Z (x; -0,125; 0; 0,125)<br>PK (x; 0;2; 4)<br>PB (x; 3,5; 4; 5; 5) | 5,1 s | $1^0$ |
| 5 | NB (x; -5; -5; -4; -3,5)<br>NK (x; -4; -3,5; -3)<br>Z (x; -3,5; 0; 3,5)<br>PK (x; 3;3,5; 4)<br>PB (x; 3,5; 4; 5; 5) | 5,2 s | $0^0$ |

Dari tabel 4.2 dapat diketahui bahwa penggunaan *membership* dari perencanaan sudah tepat karena jika dibandingkan dengan lainnya maka nilai *time seting* dan *overshoot* memiliki nilai paling optimal. Pada data no 2 memang memiliki *seting time* lebih kecil akan tetapi *overshoot* yang dimiliki termasuk cukup besar yaitu $10^0$. Sementara no 3 sampai 5 memiliki *overshoot* yang kecil bahkan sampai ada yang overshootnya $0^0$, Akan tetapi *seting time*nya besar yaitu diatas 4,8 s. Oleh karena itu *membership* dari perencanaan tidak perlu diubah karena sudah paling optimal dibandingkan 4 pengujian yang lain.

**Tabel 4.3** nilai *seting time* dan maximum *overshoot* dari perubahan *membership Output*

| No | Perubahan | *Time seting* | *Overshoot* |
|---|---|---|---|
| 1 | - | 1 s | $2^0$ |
| 2 | AMN=2,7; SDG=3,2 AMX=3,7; MX=4,2 | 9 s | $0^0$ |
| 3 | AMN=4,3; SDG=4,5 AMX=4,7; MX=4,9 | 2 s | $12^0$ |

Dari tabel diatas dapat diketahui bahwa penggunaan *output* dari perencanaan sudah tepat karena jika dibandingkan dengan lainnya maka nilai *time seting* dan *overshoot* memiliki nilai paling optimal. Pada data no 3 memang memiliki *seting time* lebih kecil akan tetapi *overshoot* yang dimiliki cukup besar yaitu $12^0$. Sementara no 2 memiliki *overshoot* $0^0$, Akan tetapi *seting time*nya besar yaitu diatas 9 s. Oleh karena itu *output* dari perencanaan tidak perlu diubah karena sudah paling optimal dibandingkan 2 pengujian yang lain.

**Tabel 4.4** nilai *time seting* dan *Max overshoot* untuk beberapa sudut

| Sudut (dari $0^0$) | *Time Seting*(Ts) | *Max Overshoot* |
|---|---|---|
| $10^0$ | 0,5 s | $1^0$ |
| $20^0$ | 1,1 s | $2^0$ |
| $30^0$ | 1,5 s | $2^0$ |
| $40^0$ | 1 s | $2^0$ |
| $50^0$ | 0,8 s | $2^0$ |
| $60^0$ | 0,8 s | $2^0$ |
| $70^0$ | 0,8 s | $1^0$ |
| $80^0$ | 1,2 s | $2,5^0$ |
| $90^0$ | 1 s | $2^0$ |
| $100^0$ | 0,9 s | $1^0$ |
| $110^0$ | 1,5 s | $2^0$ |
| $120^0$ | 1 s | $0^0$ |
| $130^0$ | 1,8 s | $1^0$ |
| $140^0$ | 1,1 s | $1^0$ |
| $150^0$ | 1,1 s | $1^0$ |
| $160^0$ | 0,8 s | $1^0$ |
| $170^0$ | 0,8 s | $1^0$ |
| $180^0$ | 1 s | $2^0$ |

| Sudut (dari $180^0$) | *Time Seting*(Ts) | *Max Overshoot* |
|---|---|---|
| $0^0$ | 0,8 s | $1^0$ |
| $10^0$ | 0,8 s | $2^0$ |
| $20^0$ | 0,7 s | $1^0$ |
| $30^0$ | 0,8 s | $2^0$ |
| $40^0$ | 1 s | $2^0$ |
| $50^0$ | 1 s | $2^0$ |
| $60^0$ | 0,6 s | $1^0$ |
| $70^0$ | 1 s | $2^0$ |
| $80^0$ | 0,8 s | $2^0$ |
| $90^0$ | 1 s | $1^0$ |
| $100^0$ | 1,3 s | $2^0$ |
| $110^0$ | 1,5 s | $2^0$ |
| $120^0$ | 3,2 s | $1^0$ |
| $130^0$ | 2,9 s | $1^0$ |
| $140^0$ | 1,9 s | $1^0$ |
| $150^0$ | 1,1 s | $1^0$ |
| $160^0$ | 1 s | $1^0$ |
| $170^0$ | 0,5 s | $1^0$ |

**Tabel 4.5** nilai *time seting* dan *Max overshoot* dari 3 jenis beban pada sudut $90^0$

| Jumlah Beban | *Time Seting* (Ts) | *Max Overshoot* |
|---|---|---|
| 1 | 1 s | $2^0$ |
| 2 | 4,5 s | $1^0$ |
| 3 | 4,5 s | $0,5^0$ |

## 5. KESIMPULAN

Setelah melakukan perencanaan dan pembuatan sistem kemudian dilakukan pengujian dan analisanya, maka dapat diambil beberapa kesimpulan tentang sistem kerja alat, yaitu sebagai berikut:

1. Sistem yang dihasilkan memiliki *time seting* 0,5 hingga 3,2 sekon dan *maximum overshoot* 0 hingga $2,5^0$
2. Pembangkit gelombang gergaji yang dibuat memiliki amplitude 5 volt dan frekuensi 50 Hz.
3. Fungsi keanggotaan dari fuzzy yang dibuat disesuaikan dengan kondisi plan yang cepat saat error besar dan melambat saat error mengecil.
4. Potensiometer sebagai sensor posisi yang digunakan sudah linier, dengan kenaikan tegangan 0,0245 volt untuk tiap derajatnya.
5. Dengan adanya beban, tercapainya *set point* juga akan semakin lama. Pada pengujian tercapainya *set point* hingga 4,5 sekon

## DAFTAR PUSTAKA


[1] *http://akbarulhuda.wordpress.com/2010/04/01/mengenal-motor-servo/*diakses 31 Oktober 2010

[2] Ringo Kusditya Nugraha. "Sistem Keamanan Rumah Berbasis Pengenalan Wicara Menggunakan DSK TMS320C6713 (Perangkat Keras)". 2009. Proyek Akhir :T. Elektronika PENS – ITS

[3] Kusumadewi, Sri dan Purnomo, Hari. 2010. *Aplikasi Logika Fuzzy untuk Pendukung Keputusan*, Yogyakarta, Indonesia.

[4] *http://www.electronic-circuitsdiagrams.com/oscillatorsimages/oscillatorsckt2.shtml* diakses 23 Desember 2010