DT-51 Application Note

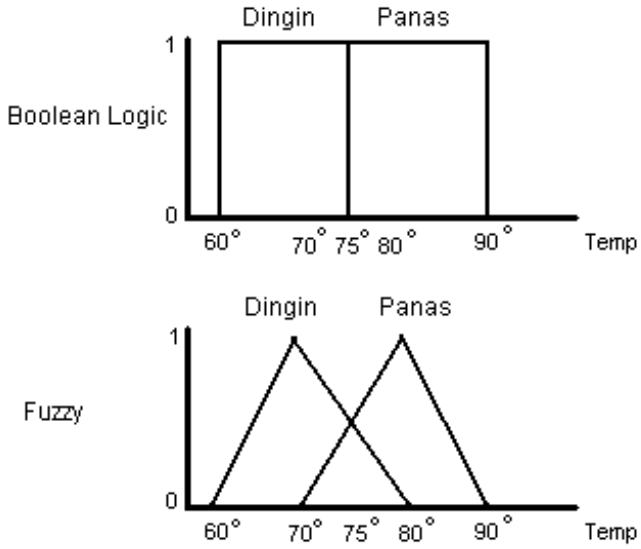
AN16 - How 2 Use DT-51 PetraFuz

oleh: Tim IE & Igit Purwahyudi (Universitas Widya Mandala)

Banyak sekali sistem kontrol pada saat ini yang menginginkan output yang mempunyai ketelitian yang tinggi, sehingga sistem kontrol tersebut membutuhkan suatu sistem yang cukup kompleks. Dengan adanya fuzzy maka maka tidak diperlukan lagi suatu sistem kontrol yang sangat kompleks.

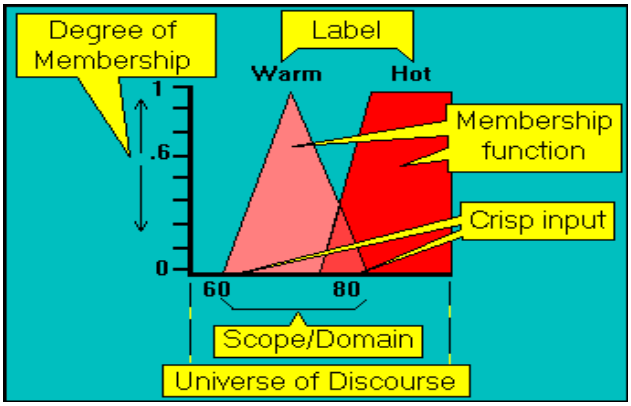
SISTEM FUZZY

Sistem Fuzzy ditemukan pertama kali oleh Prof. Lotfi Zadeh pada pertengahan tahun 1960 di Universitas California. Sistem ini diciptakan karena *boolean logic* tidak mempunyai ketelitian yang tinggi, hanya mempunyai logika 0 dan 1 saja. Sehingga untuk membuat sistem yang mempunyai ketelitian yang tinggi maka kita tidak dapat menggunakan *boolean logic*. Bedanya fuzzy dengan *boolean logic* dapat diilustrasikan pada gambar 1.



Gambar 1. Perbedaan Fuzzy Logic dan Boolean Logic

Dari contoh gambar 1, pada saat suhu berada pada 75° maka sistem yang pertama akan bingung karena batas kondisi dingin < 75 dan kondisi panas > 75°, pada fuzzy logic, suhu 75° dapat dinyatakan dengan 0.50 dingin dan 0.50 panas. Pengambilan nilai 0.50 berasal dari proses fuzzifikasi yang akan diterangkan pada proses fuzzification.



Gambar 2. Istilah yang digunakan dalam fuzzy

Pada gambar 2 dapat dilihat istilah yang digunakan dalam fuzzy dan keterangannya adalah sebagai berikut:

1. **Degree of membership**
   Fungsi dari degree of membership ini adalah untuk memberikan bobot pada suatu input yang telah kita berikan, sehingga input tadi dapat dinyatakan dengan nilai. Misalnya suhu adalah dingin, dengan adanya degree of membership maka suhu dingin tersebut dapat mempunyai suatu nilai misal 0,5. Batas dari degree of membership adalah dari 0 – 1.

2. **Scope / Domain**
   Merupakan suatu batas dari kumpulan input tertentu. Misalnya suhu dingin adalah dari 10 – 50 derajat, sangat cepat adalah dari 200 – 500 rpm.

3. **Label**
   Adalah kata – kata untuk memberikan suatu keterangan pada Scope / Domain. Contohnya : panas, dingin, cepat, sangat cepat, dll.

4. **Membership Function**
   Suatu bentuk bangun yang merepresentasikan suatu batas dari scope / domain.

5. **Crisp Input**
   Nilai input analog yang kita berikan untuk mencari degree of membership.

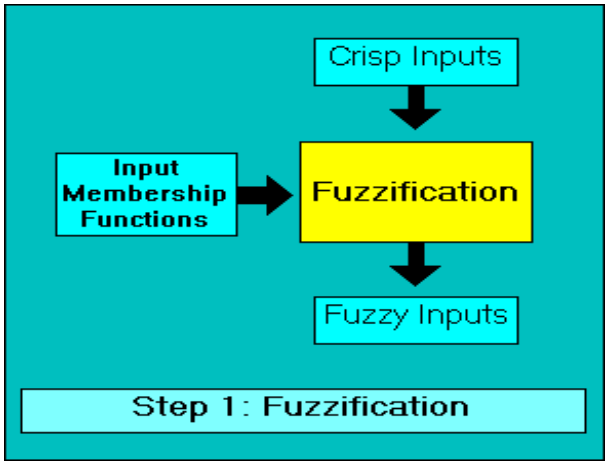
6. **Universe of discourse**
   Batas input yang telah kita berikan dalam merancang suatu fuzzy system. Batas ini berbeda dengan batas scope / domain. Universe of discourse adalah batas semua input yang akan diberikan sedangkan scope / domain adalah suatu batas yang menentukan bahwa input tersebut dinyatakan panas, dingin, cepat, dll.

   Pada fuzzy system terdapat tiga proses yaitu :

1. Fuzzification.
2. Rule evaluation.
3. Defuzification.

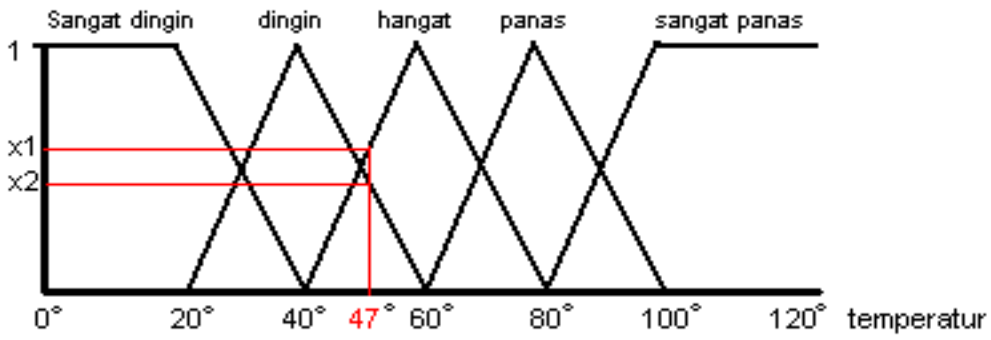
FUZZIFICATION

Proses ini berfungsi untuk merubah suatu besaran analog menjadi fuzzy input.
Secara diagram blok dapat anda lihat pada gambar 3. Prosesnya adalah sebagai berikut: suatu besaran analog dimasukkan sebagai input (crisp input), lalu input tersebut dimasukkan pada batas scope / domain sehingga input tersebut dapat dinyatakan dengan label (dingin, panas, cepat, dll) dari membership function. Membership function ini biasanya dinamakan membership function input. Dari membership function kita bisa mengetahui berapa degree of membership function-nya.
Bentuk membership function yang digunakan dalam DT-51 PetraFuz adalah bentuk trapesium dan segitiga seperti yang ditunjukkan di gambar 4.



Gambar 3. Proses Fuzzification

Contoh dari proses Fuzzification adalah seperti yang ditunjukkan di gambar 4. Sebuah sistem fuzzy untuk mengukur suhu mempunyai 5 buah membership function yang mempunyai label sangat dingin, dingin, hangat, panas, sangat panas. Kemudian input yang diperoleh dari crisp input adalah 47° maka pengambilan fuzzy input-nya adalah seperti pada gambar 4.

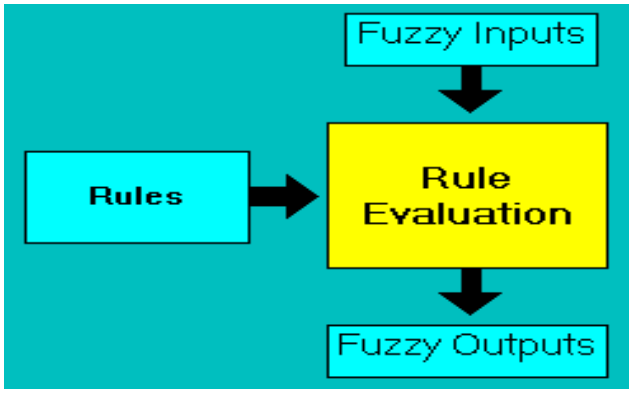


Gambar 4. Proses perubahan dari crisp input menjadi fuzzy input

Sehingga didapat 2 fuzzy input yang masing-masing adalah: dingin (x2) dan hangat (x1). Nilai x1 dan x2 dapat dicari dengan rumus persamaan garis. Yang menentukan sistem anda sensitif atau tidak adalah membership function ini. Jika membership function-nya banyak maka sistem anda menjadi sensitif. Yang dimaksud dengan sensitif dalam hal ini adalah jika input-nya berubah sedikit saja maka sistem akan cepat merespon dan menghasilkan suatu output lain. Output dari proses fuzzification ini adalah sebuah nilai input fuzzy atau yang biasanya dinamakan fuzzy input.

RULE EVALUATION

Proses ini berfungsi untuk untuk mencari suatu nilai fuzzy output dari fuzzy input.
Prosesnya adalah sebagai berikut: suatu nilai fuzzy input yang berasal dari proses fuzzification kemudian dimasukkan kedalam sebuah rule yang telah dibuat untuk dijadikan sebuah fuzzy output. Gambar diagram bloknya dapat anda lihat pada gambar 5.



Gambar 5. Diagram blok proses Rule Evaluation

Ini merupakan bagian utama dari fuzzy, karena disinilah sistem anda akan menjadi pintar atau tidak. Jika anda tidak pintar dalam mengatur rule maka sistem yang akan dikontrol menjadi kacau. Format dari rule adalah sebagai berikut:

If *antecedent1* operator *antecendent2* then *consequent1* operator *consequent2*

Contoh:

If suhu is panas and kelembaban is kering then penyemprot is sangat lama

Adanya dua *antecedent* belum tentu berarti ada dua sensor. Contohnya jika anda melakukan pengontrolan motor DC maka feedback dari motor juga dapat digunakan sebagai masukan Error sebagai input pertama dan turunan errornya atau yang biasanya disebut dError sebagai input kedua.

Ada beberapa operator yang digunakan dalam fuzzy: AND, OR, NOT. Jika operator yang digunakan adalah AND maka input terkecil yang diambil. Misalnya:

If suhu is panas (0.15) and kelembaban is kering (0.19) then penyemprot is sangat lama.

Nilai fuzzy output dari pernyataan tersebut adalah 0.15. Nilai 0.15 dan 0.19 dari contoh diatas diambil dari dua membership function input dengan cara menarik garis lurus vertikal dari nilai yang diinginkan. Seperti yang dijelaskan pada gambar 4.
Jika operator yang digunakan adalah OR maka fuzzy output-nya diambil dari nilai yang terbesar. Jika operator yang digunakan adalah operator NOT maka fuzzy output-nya adalah kebalikannya, misalnya NOT 0.9 maka akan menghasilkan 0.1 dan NOT 0.8 akan menghasilkan 0.2. Dalam melakukan perancangan dengan menggunakan DT-51 PetraFuz kita hanya menggunakan operator AND. Misalnya dari gambar 4 akan ditentukan nilai fuzzy output-nya dengan nilai fuzzy input = 10, rule – rule yang ada adalah sebagai berikut:

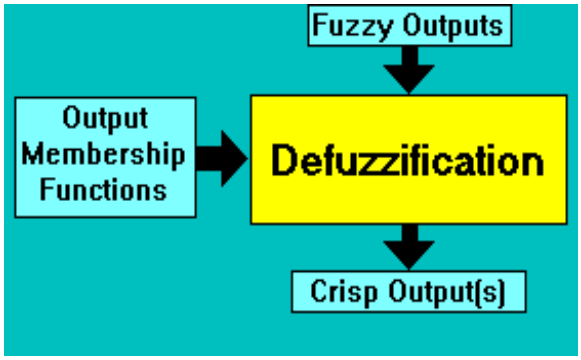
Rule:	Fuzzy output
1. if suhu is sangat dingin (1) then kipas is sangat lambat	1
2. if suhu is dingin (0) then kipas is lambat	0
3. if suhu is hangat (0) then kipas is sedang	0
4. if suhu is panas (0) then kipas is cepat	0
5. if suhu is sangat panas (0) then kipas is sangat cepat	0

Maka fuzzy output adalah: 1, 0, 0, 0, 0.

Jumlah rule maksimum = jumlah membership input 1 x jumlah membership input 2 x jumlah membership input n.

DEFUZZIFICATION

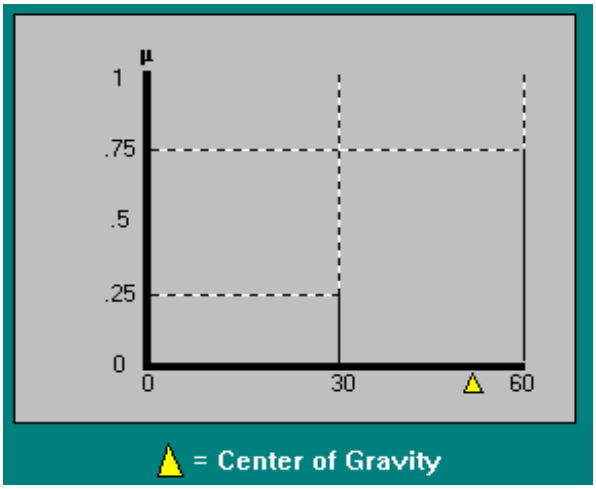
Proses ini berfungsi untuk menentukan suatu nilai crisp output.
Prosesnya adalah sebagai berikut: suatu nilai fuzzy output yang berasal dari rule evaluation diambil kemudian dimasukkan ke dalam suatu membership function output. Bentuk bangun yang digunakan dalam membership function output adalah bentuk singleton yaitu garis lurus vertikal ke atas, seperti yang ditunjukkan pada gambar 7. Besar nilai fuzzy output dinyatakan sebagai degree of membership function output. Nilai-nilai tersebut dimasukkan ke dalam suatu rumus yang dinamakan COG (Center Of Gravity) untuk mendapatkan hasil akhir yang disebut crisp output. Crisp output adalah suatu nilai analog yang akan kita butuhkan untuk mengolah data pada sistem yang telah dirancang.



Gambar 6. Proses Defuzzification

Rumus yang digunakan dalam proses ini adalah :

$$\text{Crisp Output (Y)} = \frac{\sum_i \left(\text{fuzzy output}_i\right) \times \left(\text{Singleton position on x axis}_i\right)}{\sum_i \left(\text{fuzzy output}_i\right)}$$



Gambar 7. Bentuk fuzzy output

Dengan contoh seperti gambar 7, maka perhitungan nilai crisp output-nya adalah :

$$\frac{(0) \times (0) + (.25) \times (30) + (.75) \times (60)}{0 + .25 + .75} = 52.5$$

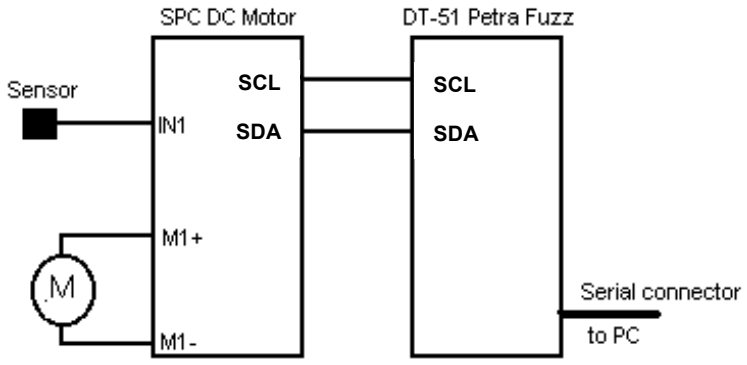
DT-51 PETRAFUZ

DT-51 PetraFuz merupakan suatu hardware yang didalamnya berisi kernel fuzzy dengan bahasa Assembly MCS-51. Sehingga kita dapat dengan mudah melakukan perancangan suatu sistem fuzzy berbasis 8051. Jumlah maksimal crisp input adalah 5 dan jumlah maksimal crisp output adalah 3. Jumlah maksimal membership function input dan output adalah 8. Untuk membership function input digunakan 4 point karena bentuknya adalah segitiga dan trapesium. Dan 1 point per output-nya. Sedangkan jumlah rule maksimum adalah 1024.
Contoh aplikasi yang akan kita rancang adalah sebuah pengaturan kecepatan motor dengan menggunakan sistem fuzzy. Hardware yang kita butuhkan adalah:

1. DT-51 PetraFuz
2. de KITS SPC DC Motor + Motor DC + Speed Encoder
3. Trafo 350 mA 9 V AC (untuk power supply DT-51 PetraFuz)
4. Beberapa kabel untuk konektor.

HARDWARE CONNECTION

Untuk koneksi tiap rangkaian dapat anda lihat pada gambar 8.



Gambar 8. Diagram blok koneksi hardware

Aplikasi ini menggunakan sebuah modul de KITS SPC DC Motor sebagai driver motor serta untuk mengontrol kecepatan motor menggunakan PWM.

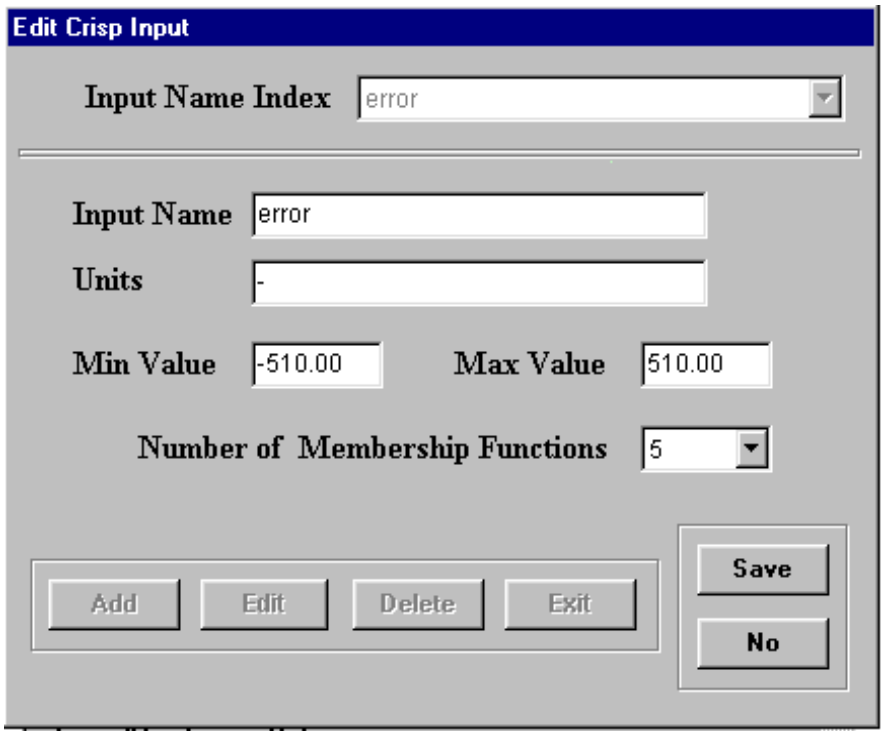
MEMASUKKAN DATA KE PETRAFUZ

Di sini kita menggunakan dua input yaitu Error (= kecepatan yang diminta - kecepatan sekarang) dan dError (= Error sekarang - Error sebelumnya) dengan nilai universe of discourse adalah –510 s/d 510. Nilai ini didapatkan dari percobaan yang telah kita lakukan. Dan nilai crisp output adalah dari –255 s/d 255.
Langkah pertama dalam merancang Fuzzy system adalah sebagai berikut:

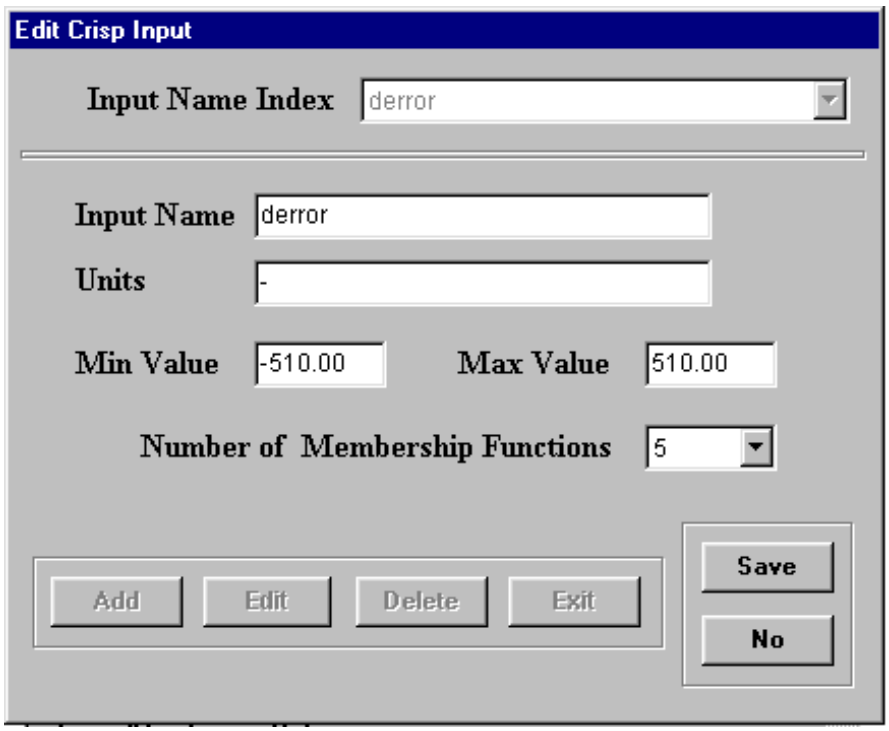
1. Merancang membership function input.
   a. Pilih menu Edit → Crisp Input.
      - Klik tombol **Add** untuk menambahkan input.
      - Setelah selesai mengisi nilai – nilai pada form yang telah tersedia maka klik tombol **Save** untuk menyimpan data.
      - Jika ingin menambah input lagi maka ulangi lagi prosesnya mulai dari penekanan tombol **Add**.

      Keterangan:
      - Input Name : Tempat anda memasukkan nama input ke-n (contoh: suhu, error)
      - Units : Satuan dari sistem anda (derajat, rpm, volt, dll)
      - Min Value : Batas bawah dari Universe of Discourse.
      - Max Value : Batas atas dari Universe of Discourse.
      - Number of membership function : Jumlah dari membership input.
      - Add : Untuk menambah input.
      - Edit : Untuk mengedit input yang telah ada.
      - Delete : Untuk menghapus input.
      - Exit : Keluar dari Form Edit Crisp Input.
      - Save : Untuk menyimpan pengisian Crisp Input.
      - No : Untuk membatalkan penyimpanan.

      Pada contoh ini nilai – nilainya dapat anda lihat pada gambar 9 dan gambar 10.



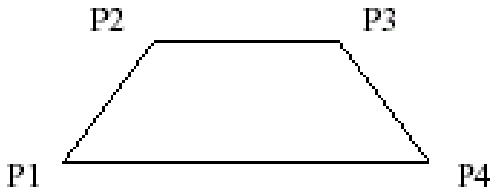
Gambar 9. Pemasukan nilai – nilai untuk input yang pertama yaitu Error



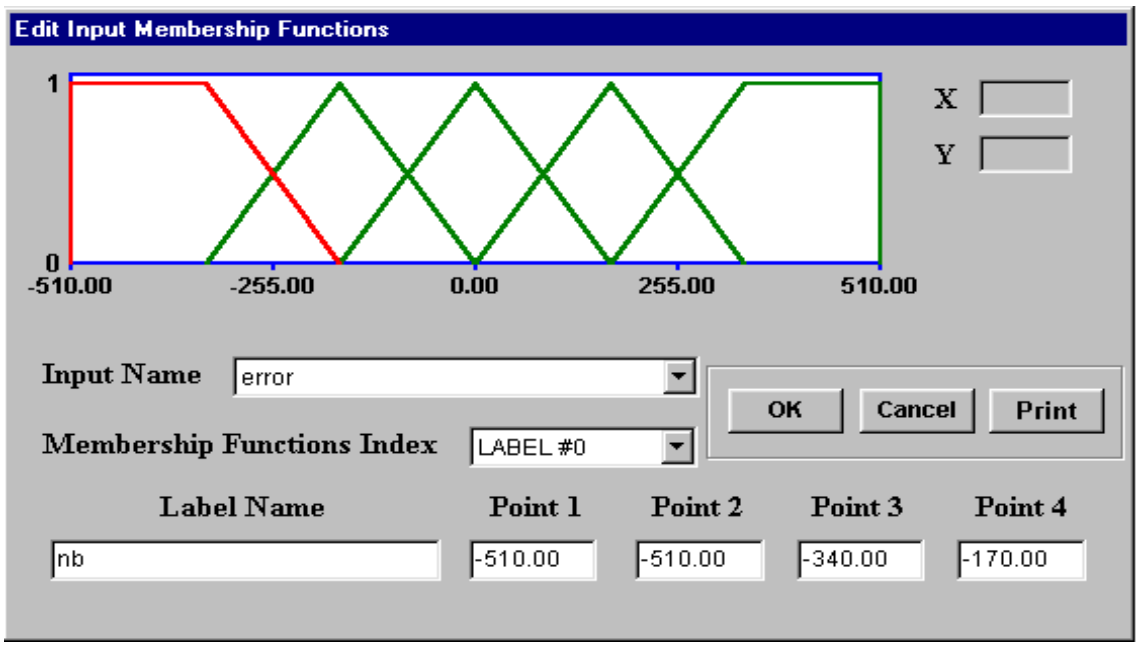
Gambar 10. Pemasukan nilai – nilai input kedua yaitu dError

b. Pilih menu Edit→ Input Membership Function.
   - Pilih salah satu input pada Input Name
   - Pilih Membership Function Index.
   - Isi nama label pada Label Name kemudian isi nilai Point1, Point2, Point3, Point4 sesuai dengan yang anda inginkan yaitu bentuk segitiga atau trapesium. Cara pemasukan nilai dapat anda lihat pada gambar 11. Jika ingin membuat bentuk segitiga maka isi Point2 = Point3.
   - Lalu ulangi langkah kedua di atas sampai pemasukan membership function input-nya selesai. Maksudnya selesai di sini adalah jumlah membership function input-nya sama dengan **Number of Membership Functions** pada form pemasukan crisp input di atas yaitu langkah a.
   - Klik tombol OK jika sudah selesai.

   Pada contoh aplikasi ini anda dapat melihat gambar 12.



Gambar 11. Cara pembuatan bangun dari membership function



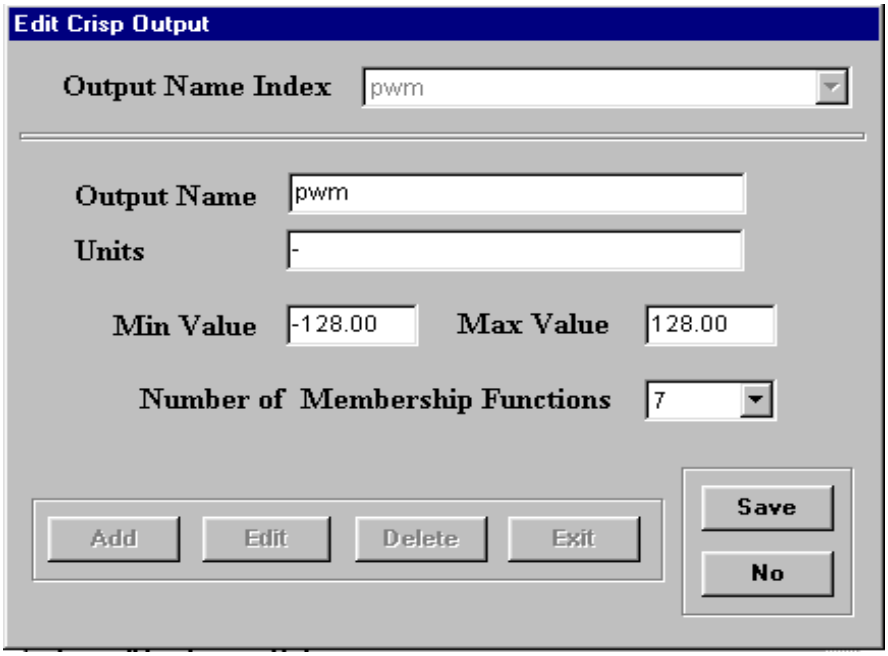
Gambar 12. Bentuk membership function untuk input Error dan input dError

Nilai-nilai yang digunakan untuk input Error dan dError dalam aplikasi pengaturan kecepatan motor adalah sebagai berikut:

Label Name	Point 1	Point 2	Point 3	Point 4
NB	-510	-510	-340	-170
NS	-340	-170	-170	0
Z	-170	0	0	170
PS	0	170	170	340
PB	170	340	510	510

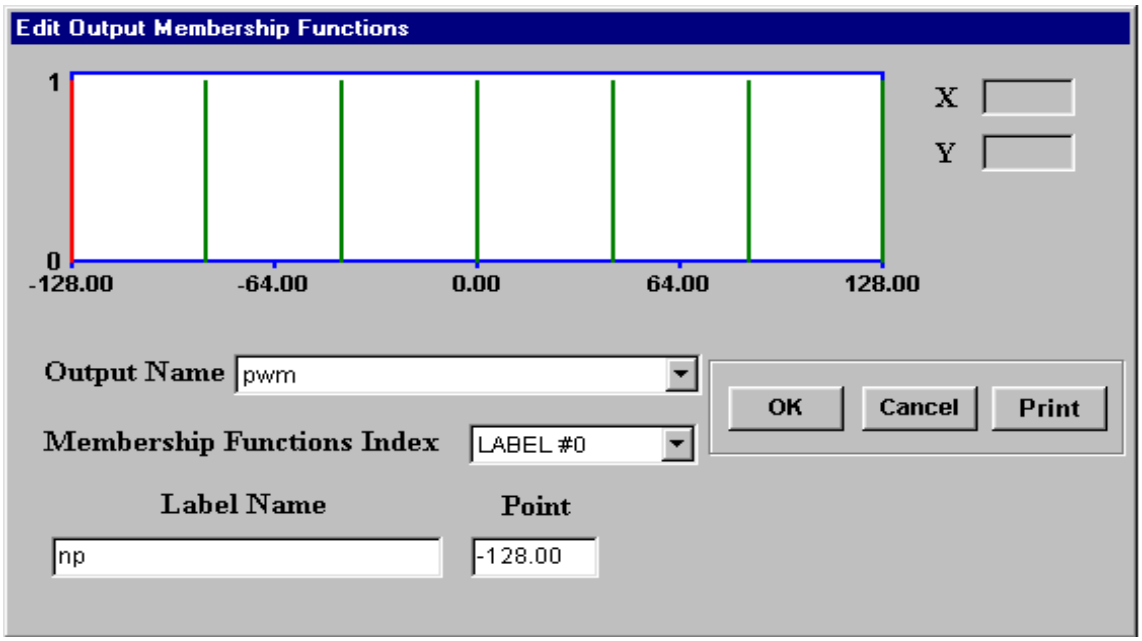
Tabel 1. Nama label membership function input dan posisi-posisi point-nya

2. Merancang membership function output-nya.
   a. Pilih menu Edit → Crisp Output
   Keterangannya sama dengan pada saat memberikan input pada langkah pemasukan nilai crisp input. Contoh dapat dilihat pada gambar 13.



Gambar 13. Proses pemasukan nilai crisp output

b. Pilih menu Edit → Output Membership Function.
Untuk membuat bentuk dari membership function output. Cara memasukkan nilai sama dengan langkah memasukkan membership function input, tetapi Point yang dimasukkan hanya satu, karena bentuk membership function output adalah singleton. Pada gambar 14 akan diperlihatkan bentuk membership function output.



Gambar 14. Bentuk dari membership function output

Nilai – nilai yang dimasukkan dalam perancangaan membership function output dalam aplikasi pengaturan kecepatan motor adalah sebagai berikut :

Label Name	Point
NB	-128
NM	-86
NS	-43
Z	0
PS	43
PM	86
PB	128

Tabel 2. Nama label membership function output dan posisi-posisi point-nya

Keterangan:
NB : Negative Big
NM : Negative Medium
NS : Negative Small
Z : Zero
PS : Positif Small
PM : Positif Medium
PB : Positif Big

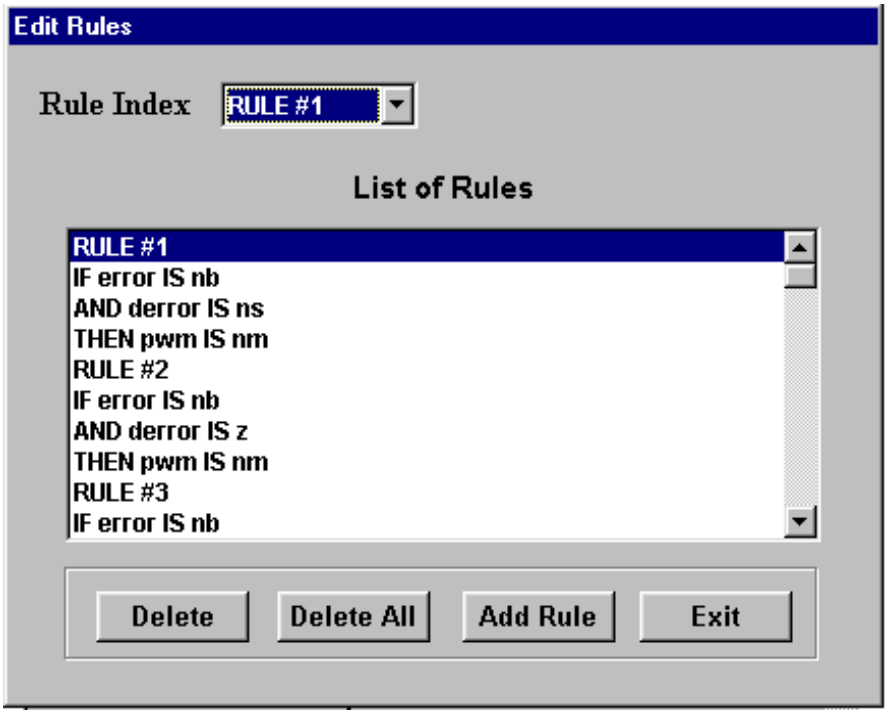
3. Merancang rule.
Pilih menu Edit → Rules.
- Untuk menambahkan rule klik tombol **Add Rule**.
- Untuk menghapus satu rule, pilih dahulu rule yang akan di hapus kemudian klik tombol **Delete**.
- Untuk menghapus semua rule klik tombol **Delete All**.

Perancangan rule dapat anda lakukan dengan menggunakan cara matriks. Contohnya seperti gambar 15.

		Error				
		NB	NS	Z	PS	PB
Derror	NB	NS	NS	PB	PS	PS
	NS	NM	NM	PM	PS	PM
	Z	NM	NB	Z	PM	PB
	PS	NB	NB	NS	PM	PB
	PB	NB	NB	NM	PB	PB

Gambar 15. Rule yang digunakan dalam aplikasi ini

Jika kondisi Error adalah NB dan dError adalah NB, berarti kecepatan sekarang jauh lebih besar dibandingkan dengan kecepatan yang diminta, tetapi ada perubahan Error sangat besar mendekati ke kondisi yang diminta. Oleh karena itu kecepatan harus diperlambat tetapi sangat kecil. Jadi untuk rule pertama adalah: if Error is NB and dError is NB then PWM is NS. Jika kondisi Errornya adalah PB dan dErrornya adalah NB, berarti kecepatan sekarang jauh lebih lambat dibandingkan dengan kecepatan yang diminta, tetapi ada perubahan Error sangat besar mendekati ke kondisi yang diminta. Oleh karena itu kecepatan harus dipercepat tetapi sangat kecil. Jadi rule berikutnya adalah: if Error is PB and dError is NB then PWM is PS. Dalam merancang suatu rule anda harus berhati – hati. Jika tidak maka sistem anda malah tidak akan berjalan sesuai dengan yang kita inginkan. Gambar 16 akan menunjukkan menu dari pembuatan rule.

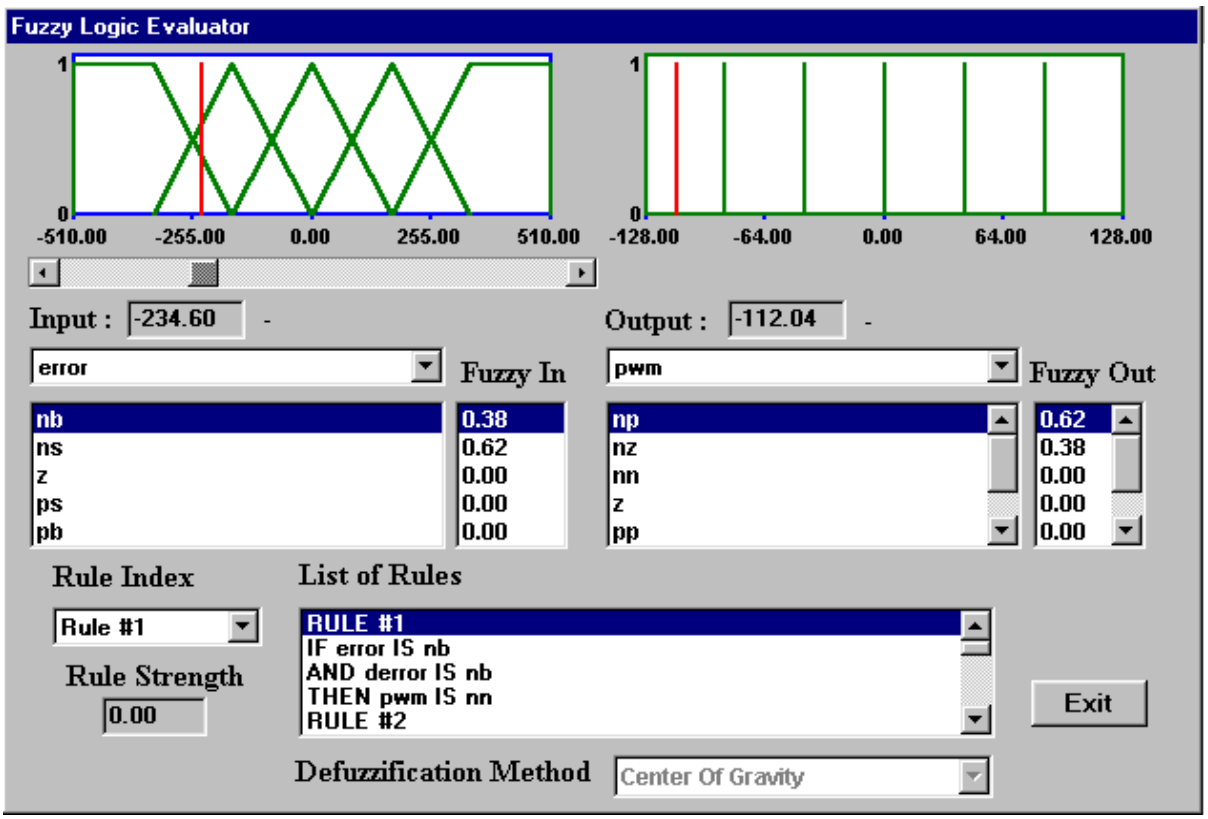


Gambar 16. Form pengisian rule pada PetraFuz

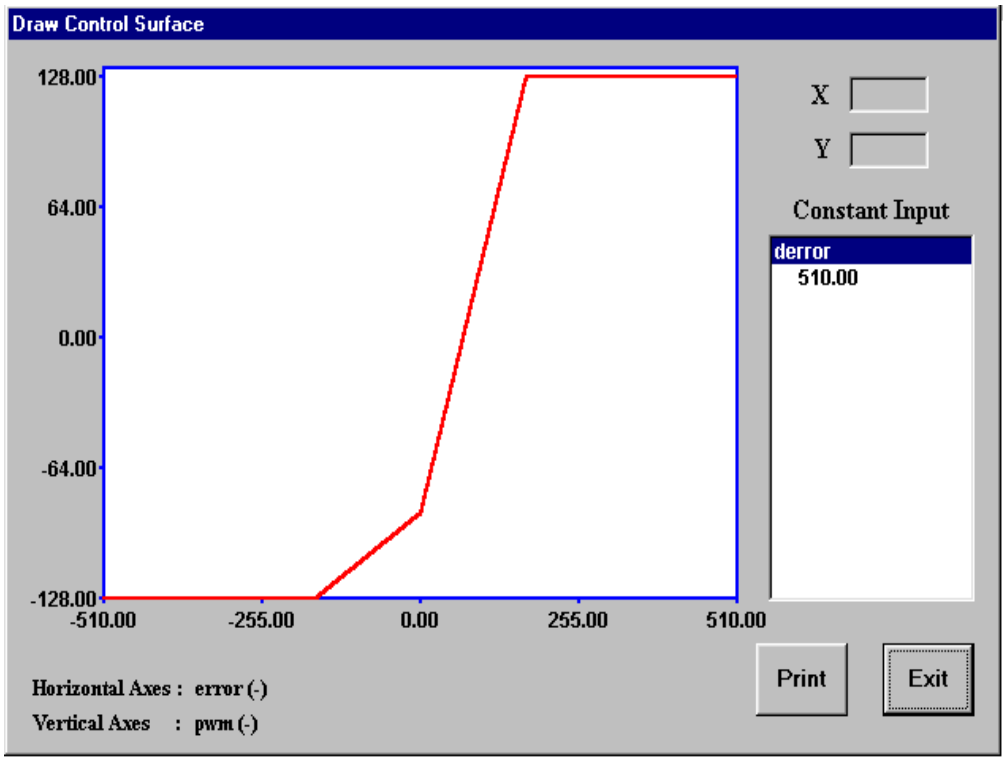
4. Mensimulasi fuzzy

   Anda dapat mensimulasi sistem anda sebelum dimasukkan ke dalam sistem yang sebenarnya. Tetapi simulasi ini tidak harus anda lakukan, karena semua nilai yang ditampilkan pada input dan output belum dinormalisasikan. Anda dapat membaca tentang normalisasi dan denormalisasi pada **Proses Normalisasi dan Denormalisasi**.

   Untuk simulasi interaktif pilih menu Evaluate → Fuzzy Logic Evaluator. Jika ingin tidak interaktif pilih menu Evaluate → Control Surface. Pada simulasi interaktif user bisa merubah nilai – nilai input dan dapat melihat nilai output pada saat terjadi perubahan. Pada simulasi tidak interaktif user hanya dapat melihat simulasi pada saat hanya satu input yang berubah dan yang satunya tetap. Contohnya dapat anda lihat pada gambar 17 untuk simulasi yang interaktif dan gambar 18 untuk simulasi yang tidak interaktif.

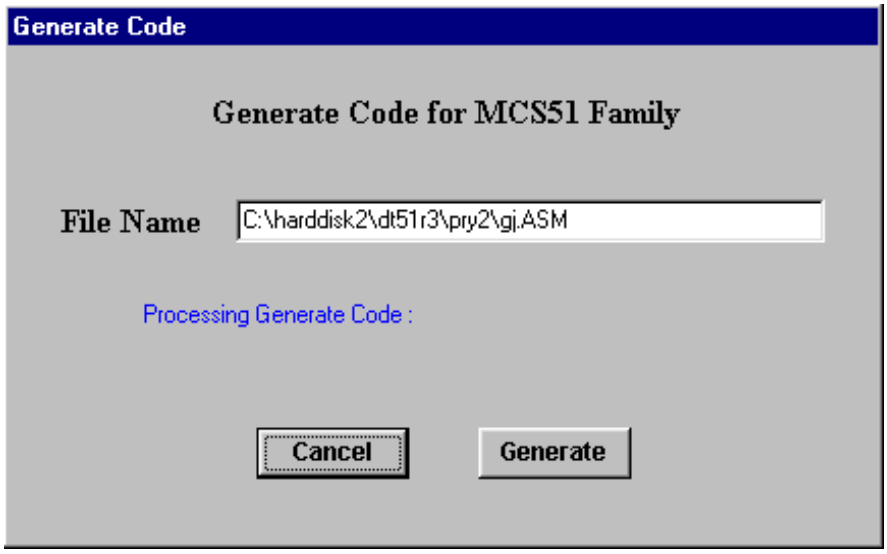


Gambar 17. Contoh simulasi interaktif



Gambar 18. Contoh simulasi tidak interaktif dengan memberikan nilai dError konstan

5. Mengkonversi fuzzy ke kode MCS-51
   Pilih menu Tools → Generate Code → MCS-51 Family. Lalu simpan dengan nama terserah anda. Lalu tekan tombol **Save**. Setelah itu akan muncul jendela Generate Code seperti yang ditunjukkan pada gambar 19. Lalu klik tombol **Generate**.



Gambar 19. Menu Generate Code.

PROSES NORMALISASI DAN DENORMALISASI

Sebelum membuat suatu program assembly dari sistem ini, proses yang terlebih dahulu dilakukan adalah menormalisasikan input dan output menjadi 0 sampai 255 karena mikrokontroler yang akan digunakan adalah mikrokontroler 8 bit. Cara menormalisasikan input dapat dilakukan dengan menggunakan persamaan garis. Contohnya jika terdapat input antara –510 s/d 510 (input sesungguhnya) dan akan dinormalisasikan menjadi 0 s/d 255 (input untuk mikrokontroler) maka penyelesaiannya dapat anda lihat pada gambar 20.

Gambar 20. Proses normalisasi crisp input

$$\frac{y - y1}{x - x1} = \frac{y2 - y1}{x2 - x1}$$

y1 = 0
y2 = 255
x1 = -510
x2 = 510

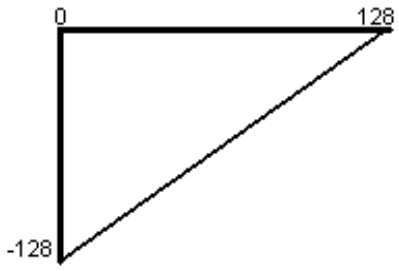
$$\frac{y - 0}{x - (-510)} = \frac{255 - 0}{510 - (-510)}$$

y = 0.25x - 127.5	rumus normalisasi

Tujuan dari proses denormalisasi adalah untuk mencari nilai crisp output sesungguhnya sehingga kita dapat mengolah data tersebut. Proses pengerjaannya sama dengan proses normalisasi. Untuk proses denormalisasi batasnya adalah 0 s/d 255 untuk koordinat x (output dari mikrokontroler) dan –128 s/d 128 untuk koordinat y (output sesungguhnya). Caranya adalah sebagai berikut: di sini akan dihitung proses denormalisasi dua kali yaitu

dari 0 s/d 128 untuk denormalisasi nilai –128 s/d 0 dan dari 129 s/d 255 untuk denormalisasi nilai 1 s/d 128. Prosesnya dapat anda lihat seperti gambar 21 dan 22.

Proses 1:



Gambar 21. Proses denormalisasi tahap 1

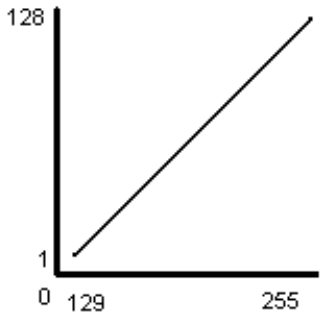
Proses pengerjaan denormalisasi tahap 1:

y1 = -128
y2 = 0
x1 = 0
x2 = 128

$$\frac{y+128}{x-0} = \frac{0+128}{128-0}$$

y = x - 128 rumus denormalisasi 1

Proses 2:



Gambar 22. Proses denormalisasi tahap 2

Proses pengerjaan denormalisasi tahap 2:

y1 = 1
y2 = 128
x1 = 129
x2 = 255

$$\frac{y-1}{x-129} = \frac{128-1}{255-129}$$

$$y = \frac{127x}{126} - \frac{16383}{126} + 1$$

rumus denormalisasi 2

PEMROGRAMAN DENGAN BAHASA ASSEMBLY

Setelah kita menemukan rumus untuk normalisasi dan denormalisasi, maka program assembly dapat dibuat. Jika anda ingin membuat suatu fuzzy system untuk aplikasi yang lain, tidak perlu harus sama dengan program yang telah ada, tetapi yang perlu diketahui sebelum membuat suatu fuzzy sistem adalah bagaimana cara untuk menormalisasikan input dan mendenormalisasikan output-nya sebelum data dapat diolah. Hal ini dilakukan agar nilai-nilai tersebut dapat diterima oleh DT-51 PetraFuz.
Routine fuzzify dari PetraFuz terletak di alamat 0900h, untuk menjalankannya anda harus menggunakan perintah LCALL 0900h. Berikut ini adalah contoh format yang digunakan dalam menggunakan DT-51 PetraFuz.

```
Fuzzify      EQU   0900H
Current_ins  EQU   0BH
Cog_Outs     EQU   0DH
```

Keterangan:
- Fuzzify : Routine PetraFuz
- Current_ins : Crisp Input PetraFuz
- Cog_Outs : Crisp Output PetraFuz

Jika anda mau memakai internal RAM maka anda harus memakai dengan alamat minimal 063H. Karena 08H – 62H digunakan oleh routine PetraFuz Engine.

Contoh program perhitungan error:

```
MOV A, SP          ; memasukkan nilai SP (kecepatan yang diminta)
MOV R0, PV         ; memasukkan nilai PV (kecepatan sekarang)
SUBB A, R0
MOV ERROR, A
```

Contoh program perhitungan dError:

```
MOV A, ERROR       ; Error(n)
MOV R0, ERROR-1    ; Error(n-1)
SUBB A, R0
MOV DERROR, A
```

Di dalam aplikasi ini yang dinormalisasikan adalah hasil dari Error dan dErrror (bukan Error dan dError), setelah itu baru dimasukkan sebagai input ke DT-51 PetraFuz. Nilai Error yang sesungguhnya harus disimpan ke dalam suatu register, karena Error yang belum dinormalisasi akan digunakan dalam perhitungan denormalisasi yaitu untuk nilai Error(n-1).

Contoh program untuk memasukkan data Error dan dError ke dalam DT-51 PetraFuz, setelah itu memanggil prosedur Fuzzify dan melihat hasilnya di register accumulator.

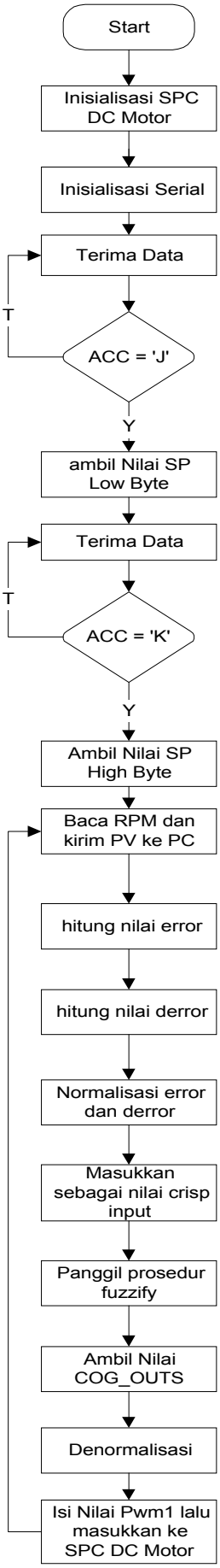
```
MOV CURRENT_INS,ERROR
MOV CURRENT_INS+1,DERROR
LCALL FUZIFY
MOV A,COG_OUTS
```

Hasil dari register accumulator di atas adalah nilai crisp output dan nilai tersebut harus didenormalisasikan.

Contoh di atas bukanlah contoh program secara khusus melainkan contoh program secara garis besar. Hal ini dikarenakan tidak adanya program untuk normalisasi dan denormalisasi. Sebelum nilai crisp input dimasukkan ke dalam DT-51 PetraFuz, nilainya harus dinormalisasikan terlebih dahulu. Nilai crisp output yang telah diperoleh di register accumulator pada contoh di atas juga belum didenormalisasikan.
Di dalam pembuatan suatu program assembly, kalau bisa hindari pembuatan program aritmatika yang cukup kompleks, misalnya perkalian atau pembagian 16 x 16 bit. Hal ini dilakukan agar proses fuzzifikasi tidak memakan banyak waktu.
Flowchart untuk aplikasi pengaturan kecepatan motor DC dengan sistem Fuzzy dapat dilihat pada gambar 23.

Proses kerjanya adalah sebagai berikut:

1. Pertama kali dilakukan proses inisialisasi de KITS SPC DC Motor dan komunikasi serial.
2. Terima data serial dari PC dan tunggu sampai karakter J dikirimkan. Lalu ambil nilai SP Low Byte.
3. Terima data serial lagi dari PC dan tunggu sampai karakter K dikirimkan, lalu ambil nilai SP High Byte.
4. Setelah itu baca nilai RPM sekarang dan kirimkan ke PC.
5. Hitung nilai Error dan dError, tetapi jangan lupa untuk menormalisasikan nilai tersebut.
6. Setelah normalisasi dilakukan maka masukkan kedua nilai yaitu Error dan dError ke dalam DT-51 PetraFuz.
7. Panggil prosedur fuzzify, lalu ambil hasilnya.
8. Setelah hasil didapatkan, maka hasil itu harus didenormalisasi sehingga akan muncul nilai crisp output yang sebenarnya.
9. Nilai crisp output kita masukkan sebagai nilai PWM untuk mengatur putaran motor DC tersebut. Ulangi langkah 4 - 9 sampai nilai RPM yang dinginkan tercapai.



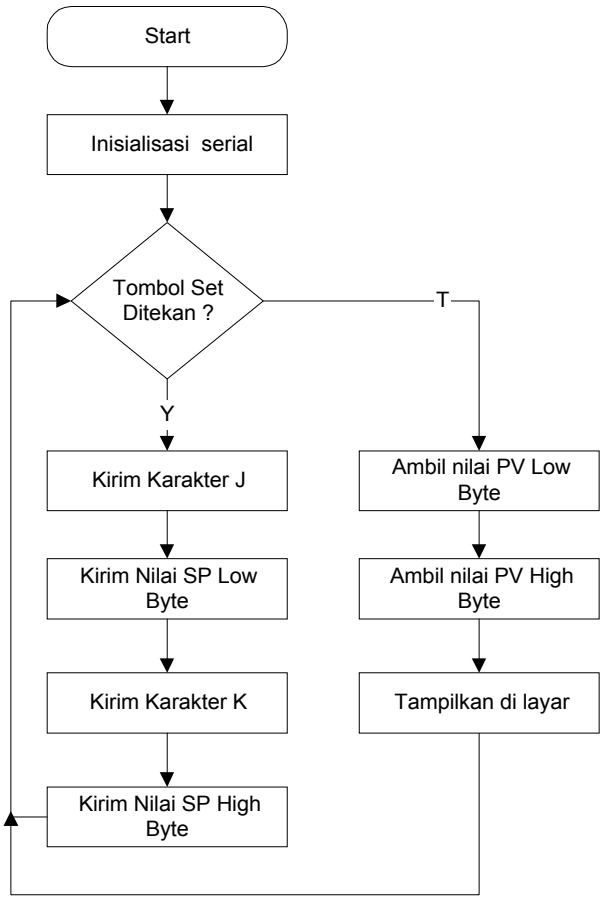
Gambar 23. Flowchart untuk program assembly pada aplikasi ini

Selain menggunakan program assembly aplikasi ini juga menggunakan sebuah program **tescoma.exe** yang dibuat dengan Delphi 5, yang berfungsi untuk komunikasi serial antara PC dengan DT-51 PetraFuz. Untuk komunikasi serial digunakan sebuah komponen Delphi yang bernama ***comport***. Program assembly harus di-download sebelum **tscoma.exe** dijalankan. Hal ini dilakukan karena program PC akan menggunakan port serial yang sama sehingga akan menghalangi proses download ke DT-51 PetraFuz.

Penjelasan cara kerjanya adalah sebagai berikut:

1. Pertama kali inisialisasi port serial dengan baudrate 9600 Bps, dan port serial diset pada COM 2.
2. Tunggu apakah tombol **Set** ditekan.
3. Jika tombol **Set** ditekan maka kirim karakter J kemudian kirim nilai SP Low Byte. Setelah itu kirim karakter K kemudian kirim nilai SP High Byte.
4. Jika tombol **Set** tidak ditekan maka ambil nilai PV low byte dan ambil nilai PV high byte.

Flowchart **tscoma.exe** dapat dilihat pada gambar 24. Contoh tampilannya dapat dilihat pada gambar 25.



Gambar 24. Flowchart untuk komunikasi serial dgn PC

Gambar 25. Tampilan program Delphi

Cara penggunaan program tersebut adalah sebagai berikut:

- Pertama kali isi nilai Set Point lalu tekan tombol **Set**.
- Jika anda ingin mengisi nilai Set Point lagi maka tekan tombol **Reset** terlebih dahulu baru setelah itu isi nilai Set Point dan kemudian tekan tombol **Set**.
- Fungsi tombol **Stop** adalah untuk menghentikan proses pengiriman data serial terakhir.
- Jika ingin keluar dari program maka tekan tombol **Close**.

TIPS PEMROGRAMAN

Pada saat menggunakan PetraFuz, ada baiknya anda melihat beberapa hal berikut ini:

1. Usahakan untuk tidak menggunakan ruang memori internal yang terlalu besar, terutama bila ingin menggunakan banyak variabel yang membutuhkan ruang memori tersendiri.
2. Jangan lupa memindah Stack Pointer jika menggunakan memori internal terutama jika menggunakan modul lain. Contohnya, jika anda menggunakan modul de KITS SPC DC Motor, di awal program Stack Pointer harus dipindah ke alamat 6DH.
3. Pada saat compiling source code dengan menggunakan ASM51 maka anda harus mengedit source yang berasal dari hasil generate PetraFuz yaitu ".CODE" dengan "CSEG". Dan semua labelnya harus diakhiri dengan “:”.

Listing program pengatur kecepatan motor dan program pembantunya terdapat di **H2UFUZ.ZIP**.

Perhatian:

File **DCMOTOR1.INC** dan **ENG_I2C1.INC** yang terdapat dalam **H2UFUZ.ZIP** merupakan **versi yang telah dimodifikasi**, **BUKAN** merupakan **versi asli** yang terdapat dalam disket de KITS SPC DC Motor.
Modifikasi dilakukan untuk memaksimalkan ruang memori yang terbatas dengan cara membebaskan sebagian alamat yang tidak digunakan agar bisa digunakan sebagai ruang untuk Stack Pointer.