ALTERNATIF IMPLEMENTASI BASIS PENGETAHUAN DALAM SISTEM BERBASIS ATURAN FUZZY

Mewati Ayub
Jurusan Teknik Informatika,
Fakultas Teknologi Informasi, Universitas Kristen Maranatha
Jl. Prof. Drg. Suria Sumantri No. 65 Bandung 40164
Email: mewati.ayub@eng.maranatha.edu

Abstract

The main elements of a fuzzy rule based system are a knowledge base and an inference engine. In this paper we present two alternatives for implementing knowledge base in a fuzzy rule based system. The first one implements knowledge base as a part of an inference engine program written in Java programming languange. The other implements knowledge base as a separate part from an inference engine program. Parser as a separate tool is needed in the later alternative to translate knowledge base in some structured semi-natural language into an intermediate format in a text file to be read efficiently by the inference engine. Comparisons between the two alternatives are discussed and some suggestions for development are given.

Keywords *: knowledge base, fuzzy rule based system, inference engine*

1. Pendahuluan

Pemanfaatan sistem berbasis aturan *fuzzy* telah meluas pada berbagai bidang dalam kehidupan sehari-hari, mulai dari peralatan rumah tangga seperti mesin cuci, mesin penanak nasi, sampai pada sistem pengatur lift, sistem pengatur lampu lalu lintas, bahkan perangkat lunak untuk diagnosis kesehatan dan sistem pembelajaran [Rez97][Ayu03][Ayu04]. Sistem berbasis aturan *fuzzy* memungkinkan penggunaan aturan linguistik untuk menggambarkan hubungan antara parameter masukan dengan keluaran yang diharapkan dari sistem yang dibangun. Aturan berbasis *fuzzy* sangat cocok diterapkan pada sistem dengan perilaku yang sulit dinyatakan dalam suatu rumusan matematis.

Penerapan aturan *fuzzy* dalam suatu sistem memerlukan mesin inferensi yang dapat menerima parameter masukan, melakukan inferensi berdasarkan aturan *fuzzy* dalam basis pengetahuan, sehingga memberikan keluaran berupa aksi yang harus dilakukan oleh sistem tersebut.

Perangkat lunak untuk mengembangkan sistem berbasis aturan *fuzzy* telah tersedia, mulai dari produk komersial sampai dengan produk *open source* [Saz00], umumnya digunakan untuk membangun aplikasi kendali *fuzzy* yang bersifat *dedicated.* Perangkat lunak tersebut menyediakan lingkungan yang memungkinkan pemakai menspesifikasikan basis pengetahuan dan bahasa pemrograman kode sumber. Spesifikasi tersebut akan dikompilasi menjadi aplikasi dengan mesin inferensi yang bersifat *dedicated.*

Mesin inferensi tidak mengandung basis pengetahuan. Mesin tersebut dapat berdiri sendiri dan melakukan inferensi berdasarkan basis pengetahuan yang diberikan. Basis pengetahuan tersebut dapat dimodifikasi untuk menala (*tuning*) aturan *fuzzy* atau diganti untuk menangani persoalan yang lain. Dalam hal ini mesin inferensi yang digunakan tidak diubah, melainkan basis pengetahuannya yang diubah [War04].

Pada makalah ini akan dibahas kedua pendekatan tersebut. Untuk pendekatan pertama, akan digunakan mesin inferensi untuk bahasa pemrograman Java, sedangkan untuk pendekatan kedua akan digunakan *parser* basis pengetahuan *fuzzy*.

2. Sistem Berbasis Aturan Fuzzy

Salah satu cara untuk merepresentasikan pengetahuan dalam bahasa semi natural adalah dengan memakai bentuk :

```
IF premis THEN konklusi
```

Bentuk `IF-THEN` tersebut seringkali disebut sebagai bentuk berbasis aturan. Apabila aturan yang dipakai merupakan aturan *fuzzy* , maka dapat dituliskan sebagai

```
IF X is A THEN Y is B
```

dengan A dan B adalah himpunan *fuzzy* [Jan97]. Dalam contoh di atas bagian premis adalah `X is A` dan bagian konklusi adalah `Y is B`.

Himpunan *fuzzy* (*fuzzy set*) A dapat didefinisikan sebagai berikut[Jan97] :

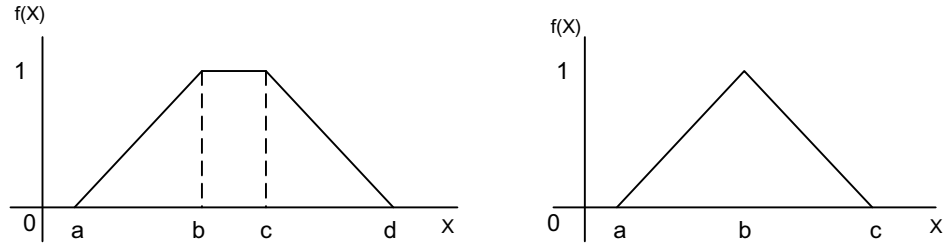
$$A = \{(x, \mu_A(x)) \mid x \varepsilon X \} \qquad (1)$$

Diberikan X adalah himpunan universal dengan x adalah elemen himpunan X. Suatu himpunan *fuzzy* A dari X ditunjukkan dengan fungsi keanggotaan $\mu_A(x)$ yang menghubungkan setiap elemen bernilai *crisp* x dengan derajat nilai keanggotaannya dalam A. Fungsi keanggotaan $\mu_A(x)$ dapat didefinisikan sebagai berikut :

$$\mu_A(x) : X \rightarrow [0,1] \qquad (2)$$

$$0 \leq \mu_A(x) \leq 1$$

Fungsi keanggotaan tersebut dapat merupakan fungsi linier maupun non linier. Beberapa fungsi keanggotaan yang umum dipakai antara lain fungsi segitiga, fungsi trapezoidal, fungsi Gaussian, dan fungsi sigmoidal [Jan97]. Pada gambar 1 ditunjukkan contoh fungsi trapezoidal dan fungsi triangular yang dispesifikasikan dengan ordinat (a,b,c,d) dan (a,b,c).



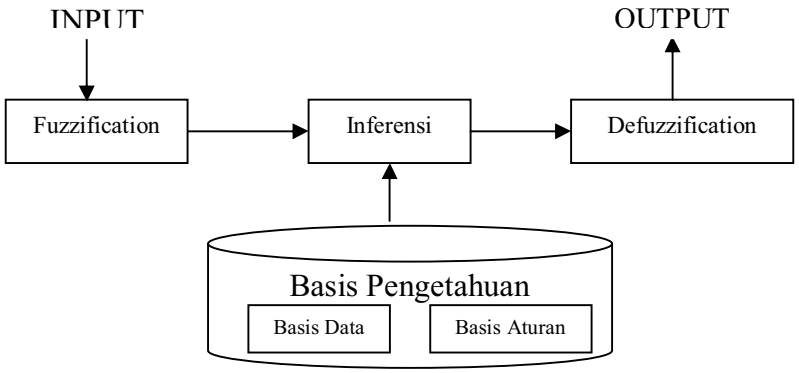
Gambar 1 Fungsi Trapezoid dan Fungsi Triangular

Pada bahasa semi natural, suatu istilah dapat diubah maknanya dengan menambahkan kata peubah di depan istilah tersebut, misalnya *very, somewhat, more, less*. Peubah ini disebut dengan *hedge*. Penambahan *hegde* di depan suatu variabel linguistik akan mengubah nilai fungsi keanggotaannya. Sebagai contoh [Ros95]:

$$"very"X = X^2 \quad (3)$$

$$"somewhat"X = \sqrt{X} \quad (4)$$

Komponen utama dalam sistem berbasis aturan *fuzzy* adalah mesin inferensi dan basis pengetahuan [Ayu03][Rez97] seperti tampak pada gambar 2. Mesin Inferensi berfungsi melakukan inferensi berdasarkan aturan *fuzzy* yang sudah didefinisikan. Basis pengetahuan terdiri atas basis data dan basis aturan. Basis data berfungsi menyediakan informasi mengenai fungsi keanggotaan, sedangkan basis aturan merupakan kumpulan aturan yang digunakan untuk proses inferensi.



Gambar 2 Sistem Berbasis Aturan *Fuzzy*

Secara umum sistem berbasis aturan *fuzzy* dapat didefinisikan sebagai suatu sistem S = (I, O, μ, R, M) dengan I adalah himpunan variabel yang merupakan masukan bagi sistem, O adalah himpunan variabel yang merupakan keluaran dari sistem, μ adalah himpunan fungsi keanggotaan untuk pemetaan nilai masukan menjadi nilai *fuzzy*, R adalah himpunan aturan *fuzzy*, dan M adalah model inferensi *fuzzy* yang dipakai oleh mesin inferensi [Ayu03].

Inferensi *fuzzy* didasarkan pada logika *fuzzy* dan melakukan penalaran seperti manusia dalam pengambilan keputusan yang mengandung ketidakpastian. Terdapat

dua model inferensi *fuzzy* yang sering dipakai yaitu model inferensi Mamdani dan model inferensi Sugeno [Jan97]. Pelaksanaan inferensi *fuzzy* pada model Mamdani melibatkan proses *fuzzification,* agregasi, komposisi dan *defuzzification* [Rez97][Dri93].

Fuzzification adalah proses untuk menentukan derajat keanggotaan nilai variabel input dalam setiap himpunan *fuzzy*. Nilai variabel input ditransformasi menjadi nilai linguistik berdasarkan fungsi keanggotaan yang telah didefinisikan. Selanjutnya akan dilakukan pencocokan nilai linguistik hasil *fuzzification* untuk setiap himpunan *fuzzy* dengan premis dalam aturan *fuzzy*. Dalam inferensi *fuzzy* dimungkinkan untuk mengaktifkan beberapa aturan yang bagian premisnya cocok dengan fakta yang diberikan.

Agregasi adalah proses perhitungan nilai dari premis aturan. Setiap premis dari aturan dinyatakan dalam derajat kebenaran yang ditentukan berdasarkan derajat keanggotaan setiap istilah linguistik. Kemudian dilakukan agregasi dengan menggunakan operasi minimum (MIN) atau produk (PROD) dari himpunan derajat kebenaran tersebut. Hasilnya merupakan derajat kebenaran dari bagian premis dalam aturan.

Komposisi adalah proses perhitungan nilai dari konklusi aturan. Penentuan derajat kebenaran dari setiap istilah linguistik dari variabel linguistik pada bagian konklusi dapat dilakukan dengan menggunakan maksimum (MAX) atau penjumlahan (SUM) dari himpunan derajat kebenaran tersebut.

Defuzzification adalah mengubah nilai linguistik dari variabel output menjadi nilai *crisp*. Salah satu teknik yang umum dipakai adalah teknik *centroid* (*centre of gravity*). Rumusnya adalah

$$COG = \frac{\int_a^b \mu_A(x)\,x\,dx}{\int_a^b \mu_A(x)\,dx} \qquad (5)$$

Bentuk aturan *fuzzy* Sugeno :

IF x is A **AND** y is B
THEN z is f(x,y)

dengan x,y adalah variabel input, z adalah variabel output, A,B adalah himpunan *fuzzy*, dan f(x,y) adalah fungsi matematis. Bagian konsekuen aturan pada inferensi Sugeno tidak mengandung himpunan *fuzzy* tapi merupakan fungsi matematis terhadap variabel input. *Defuzzification* pada inferensi Sugeno menggunakan *weighted average* dengan rumus sebagai berikut :

$$FinalOutput = \frac{\sum_{i=1}^{N} w_i z_i}{\sum_{i=1}^{N} w_i} \qquad (6)$$

3. Mesin Fuzzy untuk Java

Pada [Saz00] telah disediakan mesin *fuzzy* untuk melaksanakan inferensi *fuzzy* yang ditulis dalam bahasa pemrograman Java. Mesin *fuzzy* tersebut tersedia dalam bentuk *package* yang terdiri atas sekumpulan kelas dan metode yang memungkinkan evaluasi terhadap sekumpulan aturan *fuzzy* yang ditulis dalam bentuk simbolik.

Fitur yang disediakan mesin *fuzzy* tersebut adalah evaluasi aturan *fuzzy* dalam bentuk simbolik, operasi and/or dalam premis/konklusi aturan, ekspresi *fuzzy* bertingkat dalam premis aturan, konklusi aturan dengan beberapa variabel linguistik, *hedge* yang *predefined* dan yang didefinisikan pemakai, aturan berbobot (*weighted rules*), evaluasi aturan tunggal atau sekumpulan aturan (*rule blocks*).

Fungsi Keanggotaan yang disediakan adalah fungsi trapezoidal, spesifikasi fungsi dinyatakan dengan menyebutkan ordinat (a,b,c,d). *Hedge* yang disediakan adalah `not`,`very`, dan `somewhat`. Aturan *fuzzy* dinyatakan dalam bentuk sebagai berikut :

```
[<label>] if LV1 is [<hedge>] MF1
             [and|or LV2 is [<hedge>] MF2 [and|or ...]]
          then LVN is [<hedge>] MFN
         [and rule <label> weight is <hedge> weightMF]
```

Pada tabel 1 ditunjukkan kelas-kelas yang disediakan oleh mesin *fuzzy* untuk Java. Kelas MembershipFunction, LinguisticVariable, Hedge, HedgeNot, HedgeVery, dan HedgeSomewhat digunakan untuk mendefinisikan himpunan *fuzzy*. Sedangkan kelas FuzzyExpression, FuzzyRule, dan FuzzyBlockOfRule digunakan untuk mendefinisikan dan mengevaluasi basis aturan. Kelas EvaluationException, NoRulesFiredException, dan RulesParsingException digunakan untuk menangani kesalahan dalam evaluasi basis aturan.

Tabel 1 Daftar Kelas dalam Mesin *Fuzzy* untuk Java

No.	Nama Kelas	Deskripsi
1.	MembershipFunction	Kelas untuk menangani perhitungan fungsi keanggotaan
2.	LinguisticVariable	Kelas untuk mendefinisikan variabel linguistik
3.	Hedge	Kelas untuk menangani perhitungan *hedge*
4.	HedgeNot	Kelas untuk menangani perhitungan *hedge* not

No.	Nama Kelas	Deskripsi
5.	HedgeSomewhat	Kelas untuk menangani perhitungan *hedge* somewhat
6.	HedgeVery	Kelas untuk menangani perhitungan *hedge* very
7.	FuzzyExpression	Kelas untuk menangani ekspresi fuzzy
8.	FuzzyRule	Kelas untuk menangani aturan fuzzy
9.	FuzzyBlockOfRule	Kelas untuk mendefinisikan sekumpulan aturan fuzzy
10.	FuzzyEngine	Kelas untuk mengevaluasi satu atau sekumpulan aturan fuzzy
11.	EvaluationException	Kelas untuk menangani kesalahan dalam evaluasi aturan
12.	NoRulesFiredException	Kelas untuk menangani kesalahan jika tidak ada aturan yang dievaluasi
13.	RulesParsingException	Kelas untuk menangani kesalahan dalam *parse* aturan.

Program 1 menunjukkan contoh program Java dengan memanfaatkan mesin *fuzzy* untuk mendefinisikan sistem berbasis aturan yang digunakan pada sistem tutorial cerdas OFCL seperti telah dibahas dalam [Ayu04]. Basis aturan yang ditampilkan dalam contoh program hanyalah sebagian dari basis aturan yang digunakan OFCL.

4. Parser Basis Pengetahuan Fuzzy

Basis pengetahuan *fuzzy* berasal dari pengetahuan seorang pakar yang biasanya dispesifikasikan dalam bentuk pernyataan verbal. Untuk itu diperlukan program *parser* yang mentranslasi basis pengetahuan dalam bahasa semi natural menjadi bentuk tertentu (dalam bentuk file teks) yang lebih efisien untuk diolah mesin inferensi menjadi suatu representasi internal.

```
public class OFCL{
 public static void main(String[] args) {
 try {
  // pendefinisian fuzzy set untuk variabel masukan tsulit
  fuzzy.LinguisticVariable tsulit = new fuzzy.LinguisticVariable("tsulit");
  tsulit.add("mudah",0.0,0.0,2.0,5.5);
  tsulit.add("sedang",3.0,5.0,6.0,8.0);
  tsulit.add("sulit",5.5,8.0,10.0,10.0);
  // pendefinisian fuzzy set untuk variabel masukan nilai
  fuzzy.LinguisticVariable nilai = new fuzzy.LinguisticVariable("nilai");
  nilai.add("bawahRata",0.0,0.0,2.0,5.0);
  nilai.add("RataRata",3.0,5.0,6.0,8.0);
  nilai.add("atasRata",5.0,7.0,8.0,10.0);
  nilai.add("sangatBaik",8.0,9.5,10.0,10.0);
  // pendefinisian fuzzy set untuk variabel keluaran level
  fuzzy.LinguisticVariable level = new fuzzy.LinguisticVariable("level");
  level.add("rendah",0.0,0.0,1.0,4.0);
  level.add("memuaskan",2.0,4.0,5.0,7.0);
  level.add("tinggi",4.0,6.0,7.0,9.0);
  level.add("sangatBaik",7.0,9.0,10.0,10.0);

  //pendefinisian aturan fuzzy
  String[] Rules = new String[4];
  Rules[0] = new String("if tsulit is mudah and nilai is bawahRata
                         then level is rendah");
  Rules[1] = new String("if tsulit is mudah and nilai is RataRata
                         then level is memuaskan");
  Rules[2] = new String("if tsulit is mudah and nilai is atasRata
                         then level is tinggi");
  Rules[3] = new String("if tsulit is mudah and nilai is sangatBaik
                         then level is tinggi");

  fuzzy.FuzzyEngine FE = new fuzzy.FuzzyEngine(); //mesin fuzzy
  FE.register(tsulit);
  FE.register(nilai);
  FE.register(level);


  double val1 = 3.5; double val2 = 7.0;      //data input
  tsulit.setInputValue(val1);
  nilai.setInputValue(val2);

  fuzzy.FuzzyBlockOfRules FBR = new fuzzy.FuzzyBlockOfRules(Rules);
  FE.register(FBR);
  FBR.parseBlock();
  FBR.evaluateBlock();        //proses inferensi


  //defuzzification
  double result = level.defuzzify();
  System.out.println("Hasil Inferensi untuk input1="+val1+" input2="+val2);
  System.out.println("adalah ="+result);
  }
  catch (Exception e) { System.out.println("Ada kesalahan !");}
 }
}
```

Program 1 Contoh program yang memakai Mesin Fuzzy

Implementasi *parser* basis pengetahuan *fuzzy* memerlukan bahasa khusus untuk menspesifikasikan basis pengetahuan. Pada [War04] bahasa tersebut diberi nama *Fuzzy Knowledge Base Specification Language* (FKBSL). FKBSL terdiri atas dua bagian sebagai berikut :

1. Bagian deklarasi variabel linguistik dan fungsi keanggotaan untuk himpunan *fuzzy* yang didefinisikan
2. Bagian spesifikasi basis aturan yang meliputi deklarasi input dan output serta sekumpulan aturan *fuzzy* yang menghubungkan output dengan input dalam bahasa semi natural.

Fungsi Keanggotaan yang disediakan adalah fungsi triangular, spesifikasi fungsi dinyatakan dengan menyebutkan koordinat untuk titik (a,b,c), sedangkan *hedge* tidak disediakan.

Deklarasi basis pengetahuan akan dibaca oleh *parser* sekali jalan (*one-pass*) kemudian disimpan dalam suatu representasi internal. Proses *parse* dilakukan untuk memeriksa kesalahan sintaks dan semantik dalam deklarasi, serta memeriksa integritas dari definisi himpunan *fuzzy*. Pada akhir proses *parse*, jika tidak ada kesalahan sintaks dan semantik, *parser* akan menuliskan basis pengetahuan ke file teks dalam suatu format tertentu yang dapat dibaca oleh mesin inferensi.

Pemrosesan deklarasi variabel akan menyimpan variabel beserta himpunan *fuzzy* yang dimilikinya dalam sebuah tabel simbol. Mengingat banyaknya variabel dan banyaknya himpunan *fuzzy* dapat bervariasi maka struktur data internal yang digunakan adalah struktur data dinamik dengan pointer.

Pada saat memroses deklarasi basis aturan, setiap nama variabel dan nama himpunan *fuzzy* yang muncul dalam aturan harus diperiksa terhadap tabel simbol untuk memastikan bahwa nama tersebut sudah dideklarasi sebelumnya. Variabel yang muncul pada bagian premis aturan haruslah variabel masukan, demikian pula variabel yang muncul pada bagian konklusi aturan haruslah variabel keluaran. Untuk itu digunakan sebuah tabel Input dan sebuah tabel Output yang berisi pointer ke tabel simbol untuk variabel yang bersangkutan.

Proses yang cukup rumit adalah melakukan *parse* terhadap aturan IF-THEN. Dalam hal ini perlu dipilih struktur data yang cocok serta algoritma untuk mentranslasi aturan dalam bentuk simbolik menjadi representasi internal. [War04] menggunakan struktur pohon untuk merepresentasikan bentuk internal dari aturan. Simpul pohon dapat berupa proposisi atomik, proposisi dengan operasi `and` (and-list), dan proposisi dengan operasi `or` (or-list).

Parser basis pengetahuan *fuzzy* tersebut diimplementasikan menggunakan LEX dan YACC, yaitu perangkat untuk membangun kompailer atau interpreter [Lex06].

Berikut ini diberikan contoh deklarasi basis pengetahuan untuk sistem tutorial cerdas OFCL sesuai ketentuan sintaks dalam FKBSL.

```
VAR  tsulit
   SET  mudah  POINTS (0.0,1.0) (5.0,0.0);
   SET  sedang POINTS (2.5,0.0) (5.0,1.0) (7.5,0.0);
   SET  sulit  POINTS (5.0,0.0) (10.0,1.0);
END
VAR  nilai
   SET  bawahRata  POINTS (0.0,1.0) (5.0,0.0);
   SET  Rata-Rata  POINTS (3.0,0.0) (5.5,1.0) (8.0,0.0);
   SET  atasRata   POINTS (5.0,0.0) (7.5,1.0) (10.0,0.0);
   SET  sangatBaik POINTS (8.0,0.0) (10.0,1.0);
END
VAR  level
   SET  rendah     POINTS (0.0,1.0) (4.0,0.0);
   SET  memuaskan  POINTS (2.0,0.0) (4.5,1.0) (7.0,0.0);
   SET  tinggi     POINTS (4.0,0.0) (6.5,1.0) (9.0,0.0);
   SET  sangatBaik POINTS (7.0,0.0) (10.0,1.0);
END

RULEBASE  OFCL_Rule
INPUT tsulit, nilai;
OUPUT level;
 RULE R1
      IF tsulit IS mudah AND nilai IS bawahRata
      THEN level IS rendah ;
 RULE R2
      IF tsulit IS mudah AND nilai IS RataRata
      THEN level IS memuaskan ;
 RULE R3
      IF tsulit IS mudah AND nilai IS atasRata
      THEN level IS tinggi ;
 RULE R4
      IF tsulit IS mudah AND nilai IS sangatBaik
      THEN level IS tinggi ;
END
```

Program 2 Contoh Spesifikasi Basis Pengetahuan dalam FKBSL

5. Kesimpulan dan Saran

Terdapat dua pendekatan dalam mengimplementasikan basis pengetahuan dari suatu sistem berbasis aturan *fuzzy*. Pendekatan yang pertama adalah basis pengetahuan sebagai bagian dari program mesin inferensi *fuzzy*. Setiap kali mesin dieksekusi, spesifikasi dalam bahasa semi natural akan dibaca dan di-*parse* untuk membangun representasi internal. Representasi internal ini akan disimpan di memori dan siap untuk digunakan dalam proses inferensi selanjutnya. Pada pendekatan pertama ini proses *parse* harus diulang setiap kali mesin dieksekusi. Pendekatan pertama baik digunakan untuk pendefinisian basis aturan *fuzzy* pada tahap awal yang memerlukan modifikasi himpunan *fuzzy* ataupun modifikasi aturan. Namun hanya pemakai yang menguasai pemrograman dapat memanfaatkan pendekatan pertama ini.

Pendekatan kedua adalah dengan menyimpan basis pengetahuan secara terpisah dari program mesin inferensi. Dalam hal ini diperlukan program *parser* tersendiri untuk menterjemahkan basis pengetahuan dalam bahasa semi natural menjadi basis pengetahuan dalam suatu bentuk antara (misalnya file teks) yang siap untuk dibaca secara efisien oleh program mesin inferensi untuk disimpan dalam representasi internal. Pada pendekatan kedua, proses *parse* cukup dilakukan satu kali, karena akan menghasilkan terjemahan dalam bentuk file teks yang langsung dapat diproses

mesin inferensi. Pendekatan kedua baik digunakan apabila aturan yang dipakai sudah terdefinisi dengan baik dan tidak memerlukan perubahan. Pemakai yang tidak memahami pemrograman dapat menggunakan pendekatan kedua.

Saran pengembangan untuk pendekatan pertama adalah kemampuan untuk menggunakan beberapa fungsi keanggotaan dengan menambahkan parameter fungsi keanggotaan pada deklarasi variabel linguistik, sehingga fungsi keanggotaan dapat diperluas untuk fungsi triangular, fungsi gaussian ataupun fungsi sigmoidal

Saran pengembangan untuk pendekatan kedua adalah kemampuan untuk menggunakan beberapa fungsi keanggotaan dan *hedge*, sehingga fungsi keanggotaan dapat diperluas untuk fungsi trapezoidal, fungsi gaussian ataupun fungsi sigmoidal. Baik pada pendekatan pertama maupun kedua, dapat ditambahkan parameter model inferensi pada pendefinisian basis aturan, sehingga dapat mengakomodasi model inferensi selain inferensi Mamdani.

Daftar Pustaka

[Ayu03] *Ayub, M., Liem I.(2003). Fuzzy Control Application in Computer Assisted Learning. Proceeding CECI-SITIA di Surabaya.*

[Ayu04] *Ayub, M., The, H.L., Hinduan, A., dan Liem, I. (2004). Implementasi Kendali Fuzzy dalam Sistem Tutorial Cerdas. Proceeding Asia Pacific Conference on Art, Science, Engineering and Technology di Bandung.*

[Dri93] *Driankov, D., Hellendoorn, H., Reinfrank, M. (1993). An Introduction to Fuzzy Control. Springer-Verlag.*

[Jan97] *Jang, J.S.R, Sun, C.T., Mizutani, E. (1997). Neuro-Fuzzy and Soft Computing. Prentice Hall International.*

[Lex06] *_____ (2006). The Lex and Yacc Page. Available :* http://dinosaur.compilertools.net/#overview. *Accessed: 13/06/2006.*

[Rez97] *Reznik,L. (1997). Fuzzy Controllers. Newness.*

[Ros95] *Ross, T.J. (1995). Fuzzy Logic with Engineering Applications. Mc Graw Hill.*

[Saz00] *Sazonov, E.(2000). Fuzzy Engine for Java. Available :* www.intelligent-system.info/FuzzyEngine.htm. *Accessed: 29/05/2000.*

[War04] *Warendorf, K., Su, J.T. (2004). Application of Fuzzy Logic Techniques in The BSS1 Tutoring System. Available: http://* www.it.fht-esslingen.de/~warendorf/publications/ *FUZZY_JAIED.pdf. Accessed: 11/02/2004.*