Fuzzy Decision Tree dengan Algoritme ID3 pada Data Diabetes

F. Romansyah, I. S. Sitanggang, S. Nurdiati

***Abstract*— Decision tree is one of widely used methods in developing classification models. In order to handle uncertainty, fuzzy approach is used. This work applied a classification technique using fuzzy decision tree method on diabetes dataset to obtain classification rules for predicting new data. Fuzzy ID3 (fuzzy Iterative Dichotomiser 3) was used to develop fuzzy decision tree with high accuracy. The result is a fuzzy decision tree with the highest accuracy 94,15% for fuzziness control threshold (θ_r) = 75% and leaf decision threshold (θ_n) = 8 % or 10%.**

***Index Terms*— Fuzzy decision tree, Fuzzy ID3**

I. PENDAHULUAN

Data mining merupakan proses ekstraksi informasi atau pola penting dalam basis data berukuran besar [3]. Pada penelitian ini akan diterapkan salah satu teknik dalam *data mining*, yaitu klasifikasi terhadap data diabetes. Data diabetes yang digunakan adalah data hasil pemeriksaan lab pasien dari sebuah rumah sakit yang meliputi pemeriksaan GLUN (Glukosa Darah Puasa), GPOST (Glukosa Darah 2 Jam PP), Tg (Trigliserida), HDL (Kolesterol HDL), serta diagnosa pasien berdasarkan nilai GLUN, GPOST, HDL, dan TG.

Klasifikasi merupakan salah satu metode dalam *data mining* untuk memprediksi label kelas dari suatu *record* dalam data. Metode yang digunakan dalam penelitian ini yaitu metode klasifikasi dengan *fuzzy decision tree*. Penggunaan teknik *fuzzy* memungkinkan melakukan prediksi suatu objek yang dimiliki oleh lebih dari satu kelas. Dengan menerapkan teknik *data mining* pada data diabetes diharapkan dapat ditemukan aturan klasifikasi yang dapat digunakan untuk memprediksi potensi seseorang terserang penyakit diabetes.

Naskah diterima pada 2 Desember 2009.

F. Romansyah adalah *Associate Business Consultant* (*IT Consultant*) di PT AGIT (Astra Graphia Information Technology) 22/F ANZ Tower Jl. Jend Sudirman kavling 33A Jakarta 10220 (e-mail: fyro_mans@yahoo.com).

I. S. Sitanggang adalah staf pengajar di Departemen Ilmu Komputer FMIPA Institut Pertanian Bogor, Jl. Meranti, Wing 20 Level V, Kampus IPB Darmaga, Bogor 16680 – Indonesia (e-mail: imas.sitanggang@ipb.ac.id).

S. Nurdiati adalah staf pengajar di Departemen Ilmu Komputer FMIPA Institut Pertanian Bogor, Jl. Meranti, Wing 20 Level V, Kampus IPB Darmaga, Bogor 16680 – Indonesia (e-mail: nurdiati@ipb.ac.id)

Penelitian ini bertujuan untuk: 1) Menerapkan salah satu teknik klasifikasi yaitu *Fuzzy* ID3 (*Iterative Dichotomiser* 3) *Decision Tree* pada data hasil pemeriksaan lab pasien; 2) Menemukan aturan klasifikasi pada data diabetes yang menjelaskan dan membedakan kelas-kelas atau konsep sehingga dapat digunakan untuk memprediksi penyakit diabetes berdasarkan nilai dari atribut lain yang diketahui. Model yang dihasilkan pada penelitian ini diharapkan dapat digunakan oleh pihak yang berkepentingan untuk memprediksi potensi seseorang atau pasien terserang penyakit diabetes, sehingga terjadinya penyakit ini pada seseorang dapat diprediksi sedini mungkin dan dapat dilakukan tindakan antisipasi.

II. *FUZZY DECISION TREE*

A. Peubah Linguistik dan Linguistic Term

Peubah linguistik merupakan peubah verbal yang dapat digunakan untuk memodelkan pemikiran manusia yang diekspresikan dalam bentuk himpunan *fuzzy*. Peubah linguistik dikarakterisasi oleh *quintaple* (x, T(x), X, G, M) dengan x adalah nama peubah, T(x) adalah kumpulan dari *linguistic term*, G adalah aturan sintaks, M adalah aturan semantik yang bersesuaian dengan setiap nilai peubah linguistik. Sebagai contoh, jika umur diinterpretasikan sebagai peubah linguistik, maka himpunan dari *linguistic term* T(umur) menjadi :

T(umur) = {sangat muda, muda, tua}

Setiap *term* dalam T(umur) dikarakterisasi oleh himpunan *fuzzy*, X menunjukkan nilai interval x. Aturan semantik menunjukkan fungsi keanggotaan dari setiap nilai pada himpunan *linguistic term* [1].

Linguistic term didefinisikan sebagai kumpulan himpunan *fuzzy* yang didasarkan pada fungsi keanggotaan yang bersesuaian dengan peubah linguistik. Misalkan $\mathcal{D}$ adalah kumpulan dari *record* yang terdiri dari himpunan atribut $I = \{I_1,...,I_n\}$, dengan $I_v, v = 1,...,n$. Atribut I dapat berupa atribut numerik atau kategorikal. Untuk setiap *record d* elemen $\mathcal{D}$, $d[I_v]$ menotasikan nilai *i* dalam *record d* untuk atribut I_v. Kumpulan *linguistic term* dapat didefinisikan pada seluruh domain dari atribut kuantitatif. $\mathcal{L}_{vr}$, $r = 1,...,s_v$ menotasikan *linguistic term* yang berasosiasi dengan atribut I_v, sehingga himpunan *fuzzy* dapat didefinisikan untuk setiap $\mathcal{L}_{vr}$. Himpunan *fuzzy*, L_{vr}, $r = 1,...,s_v$ didefinisikan sebagai:

$$L_{vr} = \begin{cases} \sum dom(I_V) \dfrac{\mu_{Lvr}(i_V)}{i_V} & \text{jika } I_V \text{ diskret} \\ \int dom(I_V) \dfrac{\mu_{Lvr}(i_V)}{i_V} & \text{jika } I_V \text{ kontinu} \end{cases}$$

untuk semua $i_v \in dom(I_v)$, dengan $dom(I_v) = \{i_{v1},...,i_{vn}\}$. Derajat keanggotaan dari nilai $i_v \in dom(I_v)$ dengan beberapa *linguistic term* $\mathcal{L}_{vr}$ dinotasikan oleh $\mu_{L_{vr}}$.

B. *Fuzzy Decision Tree* (FDT)

Decision tree merupakan suatu pendekatan yang sangat populer dan praktis dalam *machine learning* untuk menyelesaikan permasalahan klasifikasi. Metode ini digunakan untuk memperkirakan nilai diskret dari fungsi target, yang mana fungsi pembelajaran direpresentasikan oleh sebuah *decision tree* [5]. *Decision tree* merupakan himpunan aturan IF…THEN. Setiap *path* dalam *tree* dihubungkan dengan sebuah aturan, di mana premis terdiri atas sekumpulan *node-node* yang ditemui, dan kesimpulan dari aturan terdiri atas kelas yang terhubung dengan *leaf* dari *path* [6].

Dalam pohon keputusan, *leaf node* diberikan sebuah label kelas. *Non-terminal node*, yang terdiri atas *root* dan *internal node* lainnya, mengandung kondisi-kondisi uji atribut untuk memisahkan *record* yang memiliki karakteristik yang berbeda. *Edge-edge* dapat dilabelkan dengan nilai-nilai *numeric-symbolic*. Sebuah atribut *numeric-symbolic* adalah sebuah atribut yang dapat bernilai *numeric* ataupun *symbolic* yang dihubungkan dengan sebuah variabel kuantitatif. Sebagai contoh, ukuran seseorang dapat dituliskan sebagai atribut *numeric-symbolic*: dengan nilai kuantitatif, dituliskan dengan "1,72 meter", ataupun sebagai nilai *numeric-symbolic* seperti "tinggi" yang berkaitan dengan suatu ukuran (*size*). Nilai-nilai seperti inilah yang menyebabkan perluasan dari *decision tree* menjadi *fuzzy decision tree* [8]. Penggunaan teknik *fuzzy* memungkinkan melakukan prediksi suatu objek yang dimiliki oleh lebih dari satu kelas.

Fuzzy decision tree memungkinkan untuk menggunakan nilai-nilai *numeric-symbolic* selama konstruksi atau saat mengklasifikasikan kasus-kasus baru. Manfaat dari teori himpunan *fuzzy* dalam *decision tree* ialah meningkatkan kemampuan dalam memahami *decision tree* ketika menggunakan atribut-atribut kuantitatif. Bahkan, dengan menggunakan teknik *fuzzy* dapat meningkatkan ketahanan saat melakukan klasifikasi kasus-kasus baru [6].

C. *Fuzzy ID3 Decision Tree*

ID3 (*Iterative Dichotomiser* 3) merupakan salah satu algoritme yang banyak digunakan untuk membuat suatu *decision tree*. Algoritme ini pertama kali diperkenalkan oleh Quinlan, menggunakan teori informasi untuk menentukan atribut mana yang paling informatif, namun ID3 sangat tidak stabil dalam melakukan penggolongan berkenaan dengan gangguan kecil pada data pelatihan. Logika *fuzzy* dapat memberikan suatu peningkatan untuk dalam melakukan penggolongan pada saat pelatihan [5].

Algoritme *fuzzy* ID3 merupakan algoritme yang efisien untuk membuat suatu *fuzzy decision tree*. Algoritme *fuzzy* ID3 adalah sebagai berikut [5]:

1. Create a *Root* node that has a set of fuzzy data with membership value 1.
2. If a node t with a fuzzy set of data *D* satisfies the following conditions, then it is a leaf node and assigned by the class name.
   - The proportion of class C_k is greater than or equal to θ_r,
     $$\frac{|D^{Ci}|}{|D|} \geq \theta_r$$
   - the number of a data set is less than θ_n
   - there are no attributes for more classifications
3. If a node *D* does no satisfy the above conditions, then it is not a leaf-node. And an new sub-node is generated as follow:
   - For A_i's (i=1,…,L) calculate the information gain, and select the test attribute A_{max} that maximizes them.
   - Devide *D* into fuzzy subset $D_1,…,D_m$ according to A_{max}, where the membership value of the data in D_j is the product of the membership value in *D* and the value of $F_{max,j}$ of the value of A_{max} in *D*.
   - Generate new node $t_i,…,t_m$ for fuzzy subsets $D_1,…,D_m$ and label the fuzzy sets $F_{max,j}$ to edges that connect between the nodes t_j and t.
   - Replace *D* by D_j (j=1,2,…,m) and repeat from 2 recursively.

D. *Fuzzy Entropy dan Information Gain*

Information gain adalah suatu nilai statistik yang digunakan untuk memilih atribut yang akan mengekspansi *tree* dan menghasilkan *node* baru pada algoritme ID3. Suatu *entropy* dipergunakan untuk mendefinisikan nilai *information gain*. *Entropy* dirumuskan sebagai berikut [5]:

$$H_s(S) = \sum_i^N - P_i * \log_2(P_i) \qquad (1)$$

dengan P_i adalah rasio dari kelas C_i pada himpunan contoh S = $\{x_1,x_2,…,x_k\}$.

$$P_i = \frac{\sum_{j=1}^{k} x_j \in C_i}{S} \qquad (2)$$

Information gain digunakan sebagai ukuran seleksi atribut, yang merupakan hasil pengurangan *entropy* dari himpunan contoh setelah membagi ukuran himpunan contoh dengan jumlah atributnya. *Information gain* didefinisikan sebagai berikut [5]:

$$G(S,A) = H(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} H(S_v) \qquad (3)$$

dengan bobot $W_i = \dfrac{|S_v|}{|S|}$ adalah rasio dari data dengan atribut *v* pada himpunan contoh.

Pada himpunan data *fuzzy*, terdapat penyesuaian rumus untuk menghitung nilai *entropy* untuk atribut dan *information*

gain karena adanya ekspresi data *fuzzy*. Berikut adalah persamaan untuk mencari nilai *fuzzy entropy* dari keseluruhan data [5]:

$$H_f(S) = H_s(S) = \sum_i^N - P_i * \log_2(P_i) \quad (4)$$

Untuk menentukan *fuzzy entropy* dan *information gain* dari suatu atribut pada algoritme *fuzzy* ID3 (FID3) digunakan persamaan sebagai berikut [5]:

$$H_f(S,A) = -\sum_{i=1}^{C} \frac{\sum_j^N \mu_{ij}}{S} \log_2 \frac{\sum_j^N \mu_{ij}}{S} \quad (5)$$

$$G_f(S) = H_f(S) - \sum_{v \subseteq A}^{N} \frac{|S_v|}{|S|} * H_f(S_v, A) \quad (6)$$

dengan μ_j adalah nilai keanggotaan dari pola ke-*j* untuk kelas ke-*i*. $H_f(S)$ menunjukkan *entropy* dari himpunan *S* dari data pelatihan pada node. $|S_v|$ adalah ukuran dari subset $S_v \subseteq S$ dari data pelatihan x_j dengan atribut *v*. $|S|$ menunjukkan ukuran dari himpunan *S*.

E. Threshold dalam Fuzzy Decision Tree

Jika pada proses *learning* dari FDT dihentikan sampai semua data contoh pada masing-masing *leaf-node* menjadi anggota sebuah kelas, akan dihasilkan akurasi yang rendah. Oleh karena itu untuk meningkatkan akurasinya, proses *learning* harus dihentikan lebih awal atau melakukan pemotongan *tree* secara umum. Untuk itu diberikan 2 (dua) *threshold* yang harus terpenuhi jika *tree* akan diekspansi, yaitu [5]:

- *Fuzziness control threshold* (FCT) / θ_r
  Jika proporsi dari himpunan data dari kelas C_k lebih besar atau sama dengan nilai *threshold* θ_r, maka hentikan ekspansi *tree*. Sebagai contoh: jika pada sebuah *sub-dataset* rasio dari kelas 1 adalah 90%, kelas 2 adalah 10% dan θ_r adalah 85%, maka hentikan ekspansi *tree*.
- *Leaf decision threshold* (LDT) / θ_n
  Jika banyaknya anggota himpunan data pada suatu *node* lebih kecil dari *threshold* θ_n, hentikan ekspansi *tree*. Sebagai contoh: sebuah himpunan data memiliki 600 contoh dengan θ_n adalah 2%. Jika jumlah data contoh pada sebuah node lebih kecil dari 12 (2% dari 600), maka hentikan ekspansi *tree*.

III. TAHAPAN PENELITIAN

Proses dasar sistem mengacu pada proses dalam *Knowledge Discovery in Database* (KDD). Proses tersebut dapat diuraikan sebagai berikut :

a. Pembersihan data, membuang data dengan nilai yang hilang dan data yang duplikat.
b. Transformasi data ke dalam bentuk data *fuzzy*.
c. Data dibagi menjadi *training set* dan test set dengan menggunakan metode *k-fold cross validation* [2].
d. Aplikasi teknik *data mining*, merupakan tahap yang penting karena pada tahap ini teknik *data mining* diaplikasikan terhadap data. Untuk menemukan aturan klasifikasi digunakan metode *fuzzy decision tree*. Langkah-langkah pada metode tersebut yaitu:
   1. Menentukan atribut yang akan digunakan.
   2. Menentukan banyaknya *fuzzy set* untuk masing-masing atribut.
   3. Menentukan banyaknya *training set* yang akan digunakan.
   4. Menghitung *membership value*.
   5. Memilih besarnya *threshold* yang akan digunakan.
   6. Membangun *fuzzy decision tree* dengan algoritme *Fuzzy* ID3.

Spesifikasi perangkat keras dan perangkat lunak yang digunakan untuk penelitian ini adalah sebagai berikut :

Perangkat keras berupa komputer personal dengan spesifikasi: Prosesor AMD Athlon 64 2800+, Memori DDR 512 MB, dan *Harddisk* 80 GB. Perangkat lunak yang digunakan adalah sistem operasi Windows XP Profesional, MATLAB 7.0 sebagai bahasa pemrograman, dan Microsoft Excel 2003.

A. Transformasi Data

Dari 5 atribut yang digunakan, 4 di antaranya merupakan atribut yang kontinu, yaitu GLUN, GPOST, HDL, dan TG, sedangkan atribut Diagnosa adalah atribut kategorik. Berdasarkan referensi hasil laboratorium, *range* normal untuk atribut GLUN, GPOST, HDL, dan TG diperlihatkan pada Tabel 1.

TABLE 1
NILAI REFERENSI HASIL LABORATORIUM

Kode Pemeriksaan	Satuan	Nilai Normal
GLUN	Mg/DL	70 ~ 100
GPOST	Mg/DL	100 ~ 140
HDL	Mg/DL	40 ~ 60
TG	Mg/DL	50 ~ 150

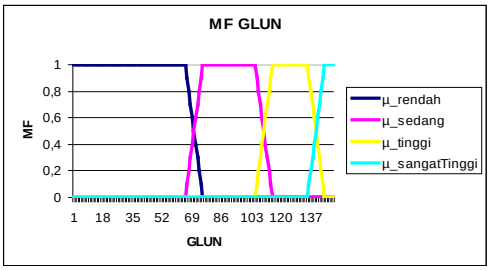
Atribut yang telah ditransformasi ke dalam bentuk *fuzzy* antara lain:

- Atribut GLUN
  Atribut GLUN dibagi menjadi 4 kelompok atau *linguistic term*, yaitu rendah (GLUN < 70 mg/DL), sedang (70 mg/DL <= GLUN < 110 mg/DL), tinggi (110 mg/DL <= GLUN < 140 mg/DL), dan sangat tinggi (GLUN >= 140 mg/DL) [4]. Dari pembagian itu dapat ditentukan *membership function* dari himpunan *fuzzy* rendah, sedang, tinggi, dan sangat tinggi untuk atribut GLUN secara terpisah yaitu:

$$\mu_{rendah}(x) = \begin{cases} 1 & ; \ x < 65 \\ \frac{x-75}{-10} & ; \ 65 <= x < 75 \\ 0 & ; \ x >= 75 \end{cases}, \quad \mu_{sedang}(x) = \begin{cases} 0 & ; \ x < 65 \\ \frac{x-65}{10} & ; \ 65 <= x < 75 \\ 1 & ; \ 75 <= x < 105 \\ \frac{x-115}{-10} & ; \ 105 <= x < 115 \\ 0 & ; \ x >= 115 \end{cases}$$

$$\mu_{tinggi}(x) = \begin{cases} 0 & ; \ x < 105 \\ \frac{x-105}{10} & ; \ 105 <= x < 115 \\ 1 & ; \ 115 <= x < 135 \\ \frac{x-145}{-10} & ; \ 135 <= x < 145 \\ 0 & ; \ x >= 145 \end{cases}, \quad \mu_{sangatTinggi}(x) = \begin{cases} 0 & ; \ x < 135 \\ \frac{x-135}{10} & ; \ 135 <= x < 145 \\ 1 & ; \ x >= 145 \end{cases}$$

Himpunan *fuzzy* untuk setiap *linguistic term* menggunakan kurva dengan bentuk trapesium seperti pada Gambar 1.



Gambar 1 Himpunan *fuzzy* atribut GLUN

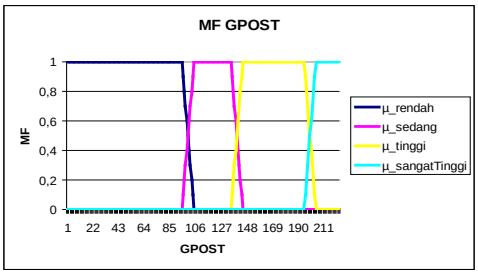
- Atribut GPOST

  Atribut GPOST dibagi menjadi 4 kelompok atau *linguistic term*, yaitu rendah (GPOST < 100 mg/DL), sedang (100 mg/DL <= GPOST < 140 mg/DL), tinggi (140 mg/DL <= GPOST < 200 mg/DL), dan sangat tinggi (GPOST >= 200 mg/DL) [4]. Dari pembagian itu dapat ditentukan *membership function* dari himpunan *fuzzy* rendah, sedang, tinggi, dan sangat tinggi untuk atribut GPOST secara terpisah yaitu:

$$\mu_{rendah}(x)=\begin{cases}1 & ; \quad x<95\\ \dfrac{x-105}{-10} & ; \quad 95<=x<105\\ 0 & ; \quad x>=105\end{cases},\quad \mu_{sedang}(x)=\begin{cases}0 & ; \quad x<95\\ \dfrac{x-95}{10} & ; \quad 95<=x<105\\ 1 & ; \quad 105<=x<135\\ \dfrac{x-145}{-10} & ; \quad 135<=x<145\\ 0 & ; \quad x>=145\end{cases}$$

$$\mu_{tinggi}(x)=\begin{cases}0 & ; \quad x<135\\ \dfrac{x-135}{10} & ; \quad 135<=x<145\\ 1 & ; \quad 145<=x<195\\ \dfrac{x-205}{-10} & ; \quad 195<=x<205\\ 0 & ; \quad x>=205\end{cases},\quad \mu_{sangatTinggi}(x)=\begin{cases}0 & ; \quad x<195\\ \dfrac{x-195}{10} & ; \quad 195<=x<205\\ 1 & ; \quad x>=205\end{cases}$$

  Himpunan *fuzzy* untuk setiap *linguistic term* menggunakan kurva dengan bentuk trapesium seperti pada Gambar 2.



Gambar 2 Himpunan *fuzzy* atribut GPOST

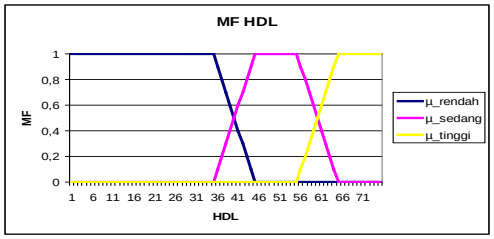
- Atribut HDL

  Atribut HDL dibagi menjadi 3 kelompok atau *linguistic term*, yaitu rendah (HDL < 40 mg/DL), sedang (40 mg/DL <= HDL < 60 mg/DL), dan tinggi (HDL >= 60 mg/DL) [4]. Dari pembagian itu dapat ditentukan *membership function* dari himpunan *fuzzy* rendah, sedang, tinggi, dan sangat tinggi untuk atribut HDL secara terpisah yaitu:

$$\mu_{rendah}(x)=\begin{cases}1 & ; \quad x<35\\ \dfrac{x-45}{-10} & ; \quad 35<=x<45\\ 0 & ; \quad x>=455\end{cases},\quad \mu_{sedang}(x)=\begin{cases}0 & ; \quad x<35\\ \dfrac{x-35}{10} & ; \quad 35<=x<45\\ 1 & ; \quad 45<=x<55\\ \dfrac{x-65}{-10} & ; \quad 55<=x<65\\ 0 & ; \quad x>=65\end{cases}$$

$$\mu_{tinggi}(x)=\begin{cases}0 & ; \quad x<55\\ \dfrac{x-55}{10} & ; \quad 55<=x<65\\ 1 & ; \quad x>=65\end{cases}$$

  Himpunan *fuzzy* untuk setiap *linguistic term* menggunakan kurva dengan bentuk trapesium seperti pada Gambar 3.



Gambar 3 Himpunan *fuzzy* atribut HDL

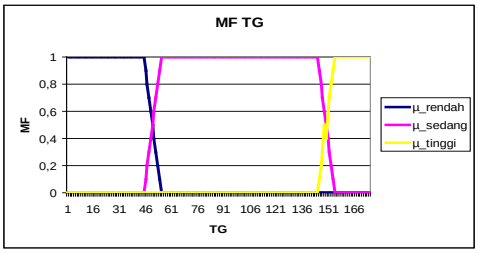
- Atribut TG

  Atribut TG dibagi menjadi 3 kelompok atau *linguistic term*, yaitu rendah (TG < 50 mg/DL), sedang (50 mg/DL <= TG < 150 mg/DL), dan tinggi (TG >= 150 mg/DL) [4]. Dari pembagian itu dapat ditentukan *membership function* dari himpunan *fuzzy* rendah, sedang, tinggi, dan sangat tinggi untuk atribut TG secara terpisah yaitu:

$$\mu_{rendah}(x)=\begin{cases}1 & ; \quad x<45\\ \dfrac{x-55}{-10} & ; \quad 45<=x<55\\ 0 & ; \quad x>=55\end{cases},\quad \mu_{sedang}(x)=\begin{cases}0 & ; \quad x<45\\ \dfrac{x-45}{10} & ; \quad 45<=x<55\\ 1 & ; \quad 55<=x<145\\ \dfrac{x-155}{-10} & ; \quad 145<=x<155\\ 0 & ; \quad x>=155\end{cases}$$

$$\mu_{tinggi}(x)=\begin{cases}0 & ; \quad x<145\\ \dfrac{x-145}{10} & ; \quad 145<=x<155\\ 1 & ; \quad x>=155\end{cases}$$

  Himpunan *fuzzy* untuk setiap *linguistic term* menggunakan kurva dengan bentuk trapesium seperti pada Gambar 4.



Gambar 4 Himpunan *fuzzy* atribut TG

Data dari atribut GLUN, GPOST, HDL, dan TG kemudian akan ditransformasi ke dalam bentuk *fuzzy* dengan menghitung derajat keanggotaan *fuzzy* pada tiap-tiap himpunan dari domain setiap atribut linguistik.

- Atribut Dignosa

  Atribut Diagnosa selanjutnya akan disebut sebagai CLASS, direpresentasikan oleh dua buah peubah linguistik yaitu ”negatif diabetes” dan ”positif diabetes”. Kedua *linguistic term*nya tersebut didefinisikan sebagai berikut:

  ”negatif diabetes” = 1

  ”positif diabetes” = 2

  Nilai atribut CLASS yang akan dikategorikan sebagai positif diabetes adalah diagnosa dengan label E10 (*Insulin-dependent diabetes mellitus*), E11 (*Non-insulin-dependet diabetes mellitus*), E12 (*Malnutrition-related diabetes mellitus*), dan E13 (*Unspecified diabetes mellitus*), selainnya dikategorikan sebagai negatif diabetes.

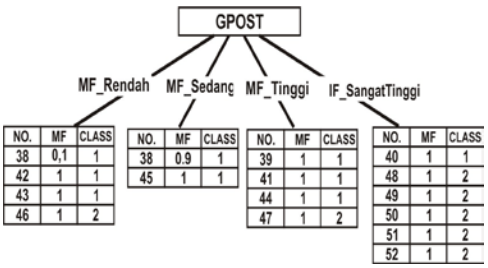
B. *Data Mining*

Pada tahap ini dilakukan teknik *data mining* menggunakan algoritme FID3 untuk membangun *fuzzy decision tree* (FDT). Data yang telah ditransformasi akan dibagi menjadi *training set* dan *test set*. Pembagian data ini menggunakan metode 10-*fold cross validation*. Data akan dibagi menjadi 10 *subset* (S_1,...,S_{10}) yang berbeda dengan jumlah yang sama besar. Setiap kali sebuah *subset* digunakan sebagai *test set* maka 9 buah partisi lainnya akan dijadikan sebagai *training set*.

Fase *training* dilakukan untuk membangun FDT dengan menggunakan algoritme FID3. Proses *training* harus melalui berbagai tahapan, sebagai contoh akan dijelaskan pembentukan FDT dengan menggunakan contoh *training set* yang terdiri dari 15 data atau *record*.

Pada contoh *training set* tersebut akan diterapkan algoritme *fuzzy* ID3 untuk menemukan model atau aturan klasifikasi. Langkah-langkah pembentukan aturan klasifikasi dengan algoritme FID3, yaitu:

1. Membuat *root node* dari semua data *training* yang ada dengan memberi nilai derajat keanggotaan untuk semua *record* sama dengan 1.
2. Menghitung *fuzzy entropy* dari *training set* yang ada. Dari hasil perhitungan diperoleh nilai *fuzzy entropy* sebesar 0.9968. Nilai ini akan digunakan untuk menghitung nilai *information gain* dari masing-masing atribut.
3. Menghitung *information gain* dari atribut GLUN, GPOST, HDL, dan TG, masing-masing diperoleh nilai 0.2064, 0.3330, 0.0304, dan 0.0050. Dari hasil tersebut dipilih atribut dengan nilai *information gain* terbesar yaitu GPOST. Atribut GPOST selanjutnya digunakan untuk mengekspansi *tree* atau menjadi *root-node*, tapi pada *sub-node* selanjutnya atribut GPOST tidak dapat digunakan untuk mengekspansi *tree*.
4. *Data training* diekspansi berdasarkan atribut GPOST sehingga diperoleh hasil seperti pada Gambar 5.



Gambar 5 Hasil ekspansi *training set* berdasarkan atribut GPOST

Nilai derajat keanggotaan yang baru masing-masing *record* pada *sub-node* diperoleh dari hasil perkalian antara derajat keanggotaan pada *root node* dan derajat keanggotaan atribut yang digunakan untuk mengekspansi *tree*. Misalkan, untuk *sub-node* dengan nilai atribut rendah, nilai derajat keanggotaan dari data no.38 $\mu_l = 0.1$ dan derajat keanggotaan dari data no.38 pada *root node* $\mu_{old} = 1$, maka nilai derajat keanggotaannya pada *sub-node* μ_{new} adalah:

$\mu_{new} = \mu_l \otimes \mu_{old} = 1 \otimes 0.1 = 0.1$

5. Menghitung proporsi dari tiap kelas yang ada pada tiap-tiap node. Misalkan, untuk *sub-node* dengan nilai keanggotaan atribut rendah, proporsi kelasnya adalah:
   $C_1 = 0.1 + 1 + 1 = 2.1$
   $C_2 = 1$
   Proporsi kelas 1 (negatif diabetes) $= \frac{C_1}{C_1 + C_2} * 100\% = 67.74\%$
   Proporsi kelas 2 (positif diabetes)
   $= \frac{C_2}{C_1 + C_2} * 100\% = 32.26\%$
6. Pada contoh ini digunakan *fuzziness control threshold* (θ_r) sebesar 80% dan *leaf decision threshold* (θ_n) sebesar 20%*15 = 3. Kedua *threshold* tersebut akan menentukan apakah *sub-node* akan terus diekspansi atau tidak. Misalkan *sub-node* dengan nilai atribut rendah yang memiliki data sebanyak 4, berdasarkan nilai proporsi kelas 1 (67.74%) dan kelas 2 (32.26%) yang lebih kecil dari θ_r (80%) dan banyaknya data atau *record* pada *sub-node* tersebut lebih besar dari θ_n, maka *sub-node* tersebut akan terus diekspansi. Lain halnya jika θ_r yang digunakan adalah 65%, maka *sub-node* tersebut tidak akan diekspansi.
7. Lakukan terus ekspansi dari *sub-node* yang ada sampai tidak ada lagi data yang dapat diekspansi atau tidak ada lagi atribut yang dapat digunakan untuk mengekspansi *tree* yaitu ketika *tree* yang terbentuk sudah mencapai kedalaman maksimum atau *sub-node* tidak memenuhi syarat dari *threshold* yang diberikan. Jika *sub-node* sudah tidak dapat diekspansi maka nilai proporsi kelas terbesar merupakan kesimpulan dari sekumpulan aturan yang diperoleh dengan menghubungkan setiap *node* yang dilewati dari *root node* hingga *leaf node*. Gambar 6 menunjukkan *fuzzy decision tree* yang diperoleh dari *training set*.

Gambar 6 *Fuzzy decision tree* untuk *training set*

IV. HASIL DAN PEMBAHASAN

Berdasarkan langkah-langkah algoritme FID3 dalam Bab III, diperoleh sebuah model yang terdiri atas 9 aturan dengan menggunakan *training set*. Model atau aturan klasifikasi yang diperoleh:

1. IF GPOST rendah AND HDL rendah THEN Negatif Diabetes
2. IF GPOST rendah AND HDL sedang THEN Negatif Diabetes
3. IF GPOST rendah AND HDL tinggi THEN Positif Diabetes
4. IF GPOST sedang THEN THEN Negatif Diabetes

5. IF GPOST tinggi AND GLUN rendah THEN Negatif Diabetes
6. IF GPOST tinggi AND GLUN sedang THEN Negatif Diabetes
7. IF GPOST tinggi AND GLUN tinggi THEN Negatif Diabetes
8. IF GPOST tinggi AND GLUN sangat tinggi THEN Positif Diabetes
9. IF GPOST sangat tinggi THEN Positif Diabetes

Berdasar metode uji 10-*fold cross validation*, proses *training* dilakukan sebanyak 240 kali. Untuk setiap *training set*, proses *training* dilakukan sebanyak 24 kali, dengan mengubah nilai θ_r sebanyak 6 kali yaitu 75%, 80%, 85%, 90%, 95%, dan 98%, dan untuk masing-masing nilai θ_r yang sama diberikan nilai θ_n yang berbeda-beda yaitu 3%, 5%, 8%, dan 10%. Rata-rata jumlah aturan yang dihasilkan pada proses *training* dan waktu eksekusi yang dibutuhkan dapat dilihat pada Tabel 2 dan Tabel 3.

TABLE 2
RATA-RATA JUMLAH ATURAN

θ_r	θ_n			
	3%	5%	8%	10%
75%	4	4	4	4
80%	7	7	7	6
85%	11	10	10	8
90%	12	11	10	8
95%	20	18	15	11
98%	27	24	20	16

Nilai-nilai θ_r dan θ_n yang digunakan dipilih berdasarkan hasil percobaan, karena dengan nilai-nilai tersebut jumlah model atau aturan yang dihasilkan mengalami perubahan yang cukup signifikan. Dengan nilai θ_r 70%, aturan yang diperoleh tidak berbeda dengan nilai θ_r 75% sehingga nilai θ_r hanya dipilih sampai 75%.

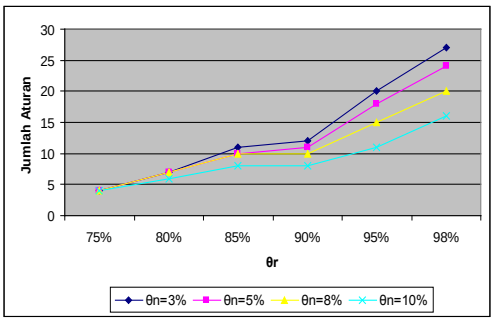
TABLE 3
RATA-RATA WAKTU EKSEKUSI DALAM DETIK

θ_r	θ_n			
	3%	5%	8%	10%
75%	0,125	0,128	0,120	0,127
80%	0,239	0,227	0,214	0,183
85%	0,344	0,334	0,308	0,244
90%	0,404	0,361	0,328	0,264
95%	0,717	0,594	0,492	0,356
98%	0,955	0,842	0,712	0,544

Berdasarkan hasil percobaan, semakin tinggi nilai θ_r semakin banyak pula aturan yang dihasilkan, hal ini terjadi karena sebelum suatu *node* didominasi oleh sebuah kelas dan proporsi untuk kelas tersebut di atas atau sama dengan nilai θ_r maka *tree* akan terus diekspansi. Pada Tabel 2 terlihat bahwa, peningkatan yang paling signifikan terjadi pada saat nilai θ_r dinaikkan dari 90% menjadi 95% dan dari 95% menjadi 98%. Kondisi semacam ini disebabkan karena pada saat ekspansi *training set* yang pertama kali dilakukan banyak *sub-node* yang proporsi salah satu kelasnya sudah mencapai nilai di atas 90%, sehingga *sub-node* tersebut tidak perlu diekspansi lagi. Ketika nilai θ_r ditingkatkan sampai 95%, baru terjadi ekspansi pada beberapa *sub-node* yang lain dan hasil proporsi kelas pada *node* di bawahnya pun ternyata banyak yang lebih rendah dari nilai θ_r yang mengakibatkan *training set* akan terus diekspansi sampai seluruh atribut terpakai.

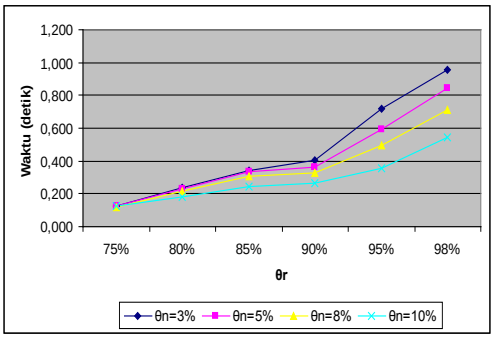
Jika diamati dengan seksama pada Tabel 2, walaupun nilai θ_n ditingkatkan, jumlah aturan yang dihasilkan tidak mengalami penurunan secara signifikan. Berdasarkan pengamatan yang dilakukan, ternyata karakteristik data pada *training set* yang digunakan tidak terlalu berbeda, pada saat terjadi ekspansi *tree* data tidak akan terlalu menyebar, karenanya jumlah himpunan data yang ada pada *sub-node* tidak berbeda jauh dengan jumlah himpunan data yang ada pada *root-node*. Dengan adanya situasi yang demikian, syarat untuk menghentikan ekspansi *tree* yaitu jumlah data atau *record* pada *sub-node* harus lebih kecil dari nilai θ_n sulit untuk tercapai.

Nilai θ_r yang terlalu rendah dan atau θ_n yang terlalu tinggi akan menghasilkan *tree* dengan ukuran yang kecil sehingga jumlah aturan yang dihasilkan juga sangat sedikit, hal ini terjadi karena *tree* yang sedang dibangun mengalami pemotongan (*pruning*) pada saat model masih mempelajari struktur dari *training set*. Sebaliknya, nilai θ_r yang terlalu tinggi dan atau θ_n yang terlalu rendah kadang kala akan menyebabkan *fuzzy decision tree* berperilaku seperti *decision tree* biasa yang tidak memerlukan adanya *threshold* sehingga menghasilkan *tree* dengan ukuran sangat besar dan jumlah aturan yang juga sangat banyak, karena *tree* akan terus diekspansi sampai *leaf-node* terdalam.



Gambar 7 Perbandingan rata-rata jumlah aturan

Gambar 7 menunjukkan perbandingan rata-rata jumlah aturan yang dihasilkan pada proses *training*. Dapat terlihat bahwa semakin tinggi nilai θ_r akan menyebabkan jumlah aturan yang dihasilkan juga meningkat, hal sebaliknya terjadi jika nilai θ_n semakin tinggi maka aturan yang dihasilkan cenderung berkurang.



Gambar 8 Perbandingan rata-rata waktu eksekusi proses *training*

Dengan melihat Gambar 7 dan Gambar 8, dapat disimpulkan bahwa, semakin tinggi nilai θ_r yang digunakan akan menghasilkan jumlah aturan yang semakin banyak sehingga waktu yang dibutuhkan untuk menghasilkan aturan-aturan tersebut juga meningkat, hal ini terjadi karena proses yang harus dilakukan untuk membangun *tree* semakin banyak.

Untuk mengukur akurasi dari model yang dihasilkan pada fase *training*, proses *testing* dilakukan sebanyak 240 kali. Proses *testing* dilakukan dengan cara memasukkan aturan yang diperoleh dari proses *training* ke dalam sebuah FIS Mamdani [7] untuk menentukan kelas dari masing-masing *record* pada *test set*. Untuk satu kali proses *training* dilakukan satu kali proses *testing*.

Dengan melihat hasil *testing*, mengubah nilai θ_r dan θ_n tidak menyebabkan adanya perubahan pada nilai akurasi dari model yang ada, walaupun jumlah aturan yang dihasilkan proses *training* mengalami peningkatan. Hal ini terjadi karena secara kebetulan semua aturan pada kedua model tersebut memiliki kelas target yang sama yaitu negatif diabetes dan *test set* yang digunakan juga berasal dari kelas target yang sama sehingga hasil *testing* dari kedua buah model dengan *test set* yang sama tidak mengalami perubahan. Namun model dengan aturan yang seragam seperti ini tidak dapat dipakai untuk melakukan prediksi karena berapapun nilai atribut yang diberikan hasilnya akan selalu negatif diabetes. Hasil *testing* akan berbeda jika aturan-aturan pada model yang dihasilkan memiliki kelas target yang berbeda. Model untuk *training set* pertama dengan θ_r (75%) dan θ_n (3%):

IF GPOST rendah THEN Negatif Diabetes
IF GPOST sedang THEN Negatif Diabetes
IF GPOST tinggi THEN Negatif Diabetes
IF GPOST sangat tinggi THEN Negatif Diabetes

Pada *test set* ketujuh dan kedelapan, nilai akurasi mengalami penurunan saat θ_r ditingkatkan menjadi 80%. Berdasarkan pengamatan yang dilakukan, keadaan seperti ini dinamakan *overfitting* karena terlalu tingginya θ_r untuk *training set* tersebut, sehingga *tree* akan terus diekspansi sampai betul-betul sesuai dengan *training set*. Akibatnya *tree* memiliki node-node yang mengandung data yang mengalami kesalahan klasifikasi.

V. EVALUASI KINERJA FID3

Evaluasi kinerja algoritme FID3 dapat diketahui dengan cara menghitung rata-rata akurasi dari seluruh proses *testing* pada 10 *test set* yang berbeda. Evaluasi kinerja dari algoritme FID3 pada nilai θ_r dan θ_n yang berbeda dapat dilihat pada Tabel 4.

TABLE 4
AKURASI *FUZZY DECISION TREE*

θ_r	θ_n			
	3%	**5%**	**8%**	**10%**
75%	94,14	94,14	94,15	94,15
80%	92,07	92,07	93,45	93,45
85%	92,07	92,07	93,45	93,45
90%	92,07	92,07	93,45	93,45
95%	90,69	91,73	93,10	93,45
98%	90,69	91,73	93,10	93,45

Dari Tabel 4 dapat dilihat bahwa kinerja algoritme FID3 mengalami penurunan jika nilai θ_r semakin besar dan atau nilai θ_n semakin kecil, walaupun penurunan yang terjadi tidaklah signifikan sehingga masih dapat ditoleransi. Seperti telah dijelaskan sebelumnya kondisi ini disebabkan karena terjadinya fenomena *overfitting*. Nilai akurasi terbaik yaitu 94,15% diperoleh pada θ_r = 75% dan θ_n = 8 % atau 10%.

VI. KESIMPULAN

Dari berbagai percobaan yang dilakukan terhadap data Diabetes didapat kesimpulan bahwa algoritme FID3 memiliki kinerja yang baik dalam membentuk *fuzzy decision tree* untuk data Diabetes yang ada. Nilai akurasi terbaik dari model yaitu 94,15% diperoleh pada *fuzziness control threshold* (θ_r) = 75% dan *leaf decision threshold* (θ_n) = 8 % atau 10%. Nilai θ_r dan θ_n sangat berpengaruh terhadap jumlah aturan yang dihasilkan, nilai θ_r yang terlalu tinggi akan menyebabkan turunnya nilai akurasi. Di lain pihak, nilai θ_n yang terlalu rendah juga dapat menyebabkan akurasi menurun.

REFERENCES

[1] E Cox. *Fuzzy Modeling and Algorithms for Data mining and Exploration*. USA: Academic Press. 2005.
[2] L. Fu. *Neural Network In Computer Intellegence*. Singapura: McGraw Hill. 1994.
[3] J. Han and M. Kamber. *Data Mining Concepts and Techniques*. Simon Fraser University. USA: Morgan Kaufman, 2006
[4] Herwanto. *Pengembangan Sistem Data Mining untuk Diagnosis Penyakit Diabetes Menggunakan Algoritme Classification Based Association* [Tesis]. Bogor. Departemen Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Institut Pertanian Bogor, 2006.
[5] G. Liang. *A Comparative Study of Three Decision Tree algorithms: ID3, Fuzzy ID3 and Probabilistic Fuzzy ID3*. Informatics & Economics Erasmus University Rotterdam Rotterdam, the Netherlands, 2005.
[6] C. Marsala. *Application of Fuzzy Rule Induction to Data Mining*. University Pierre et Marie Curie. 1998.
[7] Ormos L. *Soft Computing Method On Thom's Catastrophe Theory For Controlling Of Large-Scale Systems*. University of Miskolc, Department of Automation. 2004.
[8] Y. Yuan dan Shaw M J. *Induction of fuzzy decision trees, Fuzzy Sets and Systems* Vol. 69. 1995.

Firat Romansyah dilahirkan di Indonesia pada tahun 1985. Dia memperoleh gelar Sarjana Ilmu Komputer dari Departemen Ilmu Komputer, Institut Pertanian Bogor, Indonesia pada tahun 2007. Saat ini dia bekerja sebagai *Associate Business Consultant* (*IT Consultant*) di PT AGIT (Astra Graphia Information Technology).

Imas Sukaesih Sitanggang dilahirkan di Indonesia pada tahun 1975. Dia memperoleh gelar Sarjana Matematika dari Departemen Matematika, Institut Pertanian Bogor, Indonesia pada tahun 1997 dan mendapat gelar Master Ilmu Komputer dari Universitas Gadjah Mada, Indonesia pada tahun 2002. Saat ini dia bekerja sebagai dosen di Departemen Ilmu Komputer, Institut Pertanian Bogor, Indonesia. Topik utama penelitiannya adalah dalam *spatial data mining*.

Sri Nurdiati dilahirkan di Indonesia pada tahun 1960. Dia memperoleh gelar Sarjana Statistika dari Institut Pertanian Bogor, Indonesia pada tahun 1984 dan mendapat gelar Master Ilmu Komputer dari *The University of Western Ontario* di Canada pada tahun 1991. Pada tahun 2005 dia memperoleh gelar Doktor Matematika Terapan dari *Twente University* di Belanda. Saat ini dia bekerja sebagai dosen di Departemen Ilmu Komputer, Institut Pertanian Bogor, Indonesia. Topik utama penelitiannya adalah *scientific computing*.

.