**LECTURE NOTES**

**Selected Topics in Computational Intelligence I**

**Session 10**

**Adaptive neuro-fuzzy inference system (ANFIS)**

**LEARNING OUTCOMES**

**Tujuan Instruksional Umum :**

1. *Mahasiswa mampu menjelaskan tentang Adaptive Neuro Fuzzy Inference Systems(ANFIS)*

**Tujuan Instruksional Khusus :**

1. *Mahasiswa dapat menyebutkan sejarah ANFIS*
2. *Mahasiswa mampu menjelaskan tentang konsep pembelajaran ANFIS*
3. *Mahasiswa mampu membuat aplikasi sederhana yang menerapkan ANFIS*

**10.1 Pengenalan ANFIS**

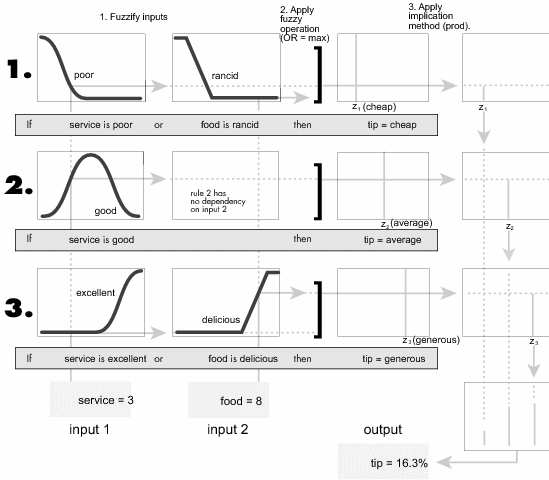
Soft computing memainkan peranan penting bagi dunia robotika saat ini. Prof. Lotfi A. Zadeh adalah seorang pakar kecerdasan buatan terkemuka yang mengusulkan istilah soft computing. Beliau mendefinisikan Soft computing sebagai penggabungan Fuzzy logic, Neuro Computing dan Evolutionary Computation dan Probabilistic Computing ke dalam suatu bidang multidisiplin[1]. Tujuan utama soft computing ialah terbentuknya high machine intelligence quotient, yaitu suatu sistem yang mampu mengolah informasi secara cerdas. SC memiliki toleransi terhadap ketidakakuratan, ketidakpastian dan kebenaran parsial untuk mencapai robustness.

Pada dasarnya, teknik Soft Computing terdiri dari Fuzzy logic, neuro computing atau Artificial neural network (ANN), Evolutionary Computation dan probabilistic computing. Pada bab ini soft computing yang dibahas akan difokuskan ke Neural network dan ANFIS karena cukup banyak digunakan pada navigasi robot cerdas. ANFIS ini merupakan gambungan dari kehebatan logika fuzzy dan neural networks. Logika fuzzy juga dapat diterapkan pada sistem kontrol pada robot. Logika Fuzzy (logika samar) merupakan logika yang berhadapan langsung dengan konsep kebenaran sebagian, dimana logika klasik menyatakan bahwa segala hal dapat diekspresikan  dalam binary 0 atau 1. Logika fuzzy memungkinkan nilai keanggotaan antara 0 dan 1.

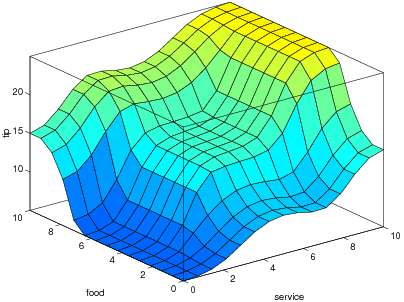
Sebagai contoh, gambar di bawah menampilkan fuzzy tipping model untuk menggambarkan sistem Sugeno. Kasus yang umum dan sederhana serta mudah dipahami jika kita ingin memberikan uang tips kepada seorang pelayan restoran berdasarkan pelayanan yang diberikan dan rasa masakan, dapat dicoba dengan kode MATLAB berikut :

a = readfis('tippersg');

gensurf(a)



**Gambar 10.1 Contoh fuzzy untuk tips**

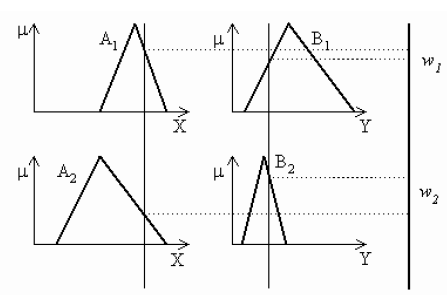


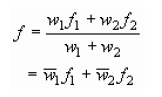
**Gambar 10.2 Hasil pembangkitan grafik untuk kasus tipper**

Terlihat dari gambar diatas, pemberian tips sangat mulus dan tidak kaku. ANFIS ialah penggabungan mekanisme Fuzzy Inference system yang digambarkan dalam arsitektur neural networks. Metode ini jauh lebih cepat dibandingkan backpropagation pada neural networks, dengan hanya beberapa puluh epoch saja, proses pembelajaran dapat selesai, basis aturan yang digunakan ialah :

Basis aturan :

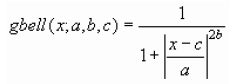
* If x is A1 and Y is B1 then z1 = p1\*x + q1\*y + r1
* If x is A2 and Y is B2 then z2=p2\*x + q2\*y + r2



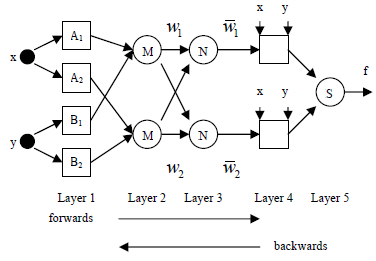
 (3)

**Gambar 10.3 output ANFIS**

Biasanya fungsi keanggotaan fuzzy input (premis) yang digunakan ialah fungsi generalized-Bell yang dirumuskan:

 (4)

Salah satu bentuk struktur yang sudah sangat dikenal dari sistem ANFIS seperti terlihat pada Gambar 12.4. Dalam struktur ini, sistem inferensi fuzzy yang diterapkan adalah inferensi fuzzy model Takagi-Sugeno-Kang.



**Gambar 10.4 Struktur ANFIS[2]**

Seperti terlihat pada gambar, sistem ANFIS terdiri dari 5 lapisan, lapisan yang disimbolkan dengan kotak adalah lapisan yang bersifat adaptif. Sedangkan yang disimbolkan dengan lingkaran adalah bersifat tetap. Setiap keluaran dari masing-masing lapisan disimbolkan dengan Ol,i dengan  i adalah urutan simpul dan l adalah menunjukan urutan lapisannya. Berikut ini adalah penjelasan untuk setiap lapisan, yaitu:

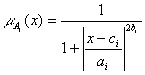
**Lapisan 1:**

Berfungsi untuk membangkitkan derajat keanggotaan :

http://trensains.com/neuro_fuzz_files/image002.gif

http://trensains.com/neuro_fuzz_files/image004.gif

Dengan x dan y adalah masukan bagi simpul ke-i:



dengan {ai, bi dan ci} ialah parameter dari fungsi keanggotaan atau disebut sebagai parameter premise.

**Lapisan 2 :**

Berfungsi untuk membangkitkan firing-strength dengan mengalikan setiap sinyal masukan.

http://trensains.com/neuro_fuzz_files/image008.gif

**Lapisan 3:**

Menormalkan *firing strength :*

http://trensains.com/neuro_fuzz_files/image010.gif

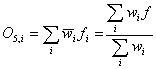
**Lapisan 4:**

Menghitung keluaran kaidah berdasarkan parameter consequent {pi, qi dan ri}

http://trensains.com/neuro_fuzz_files/image012.gif

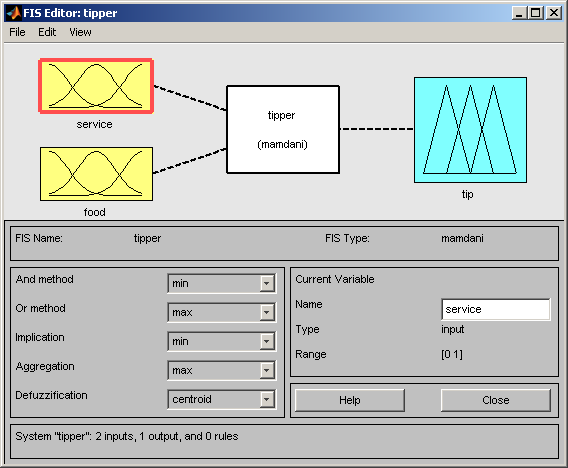
**Lapisan 5**

Menghitung sinyal keluaran ANFIS dengan menjumlahkan semua sinyal yang masuk



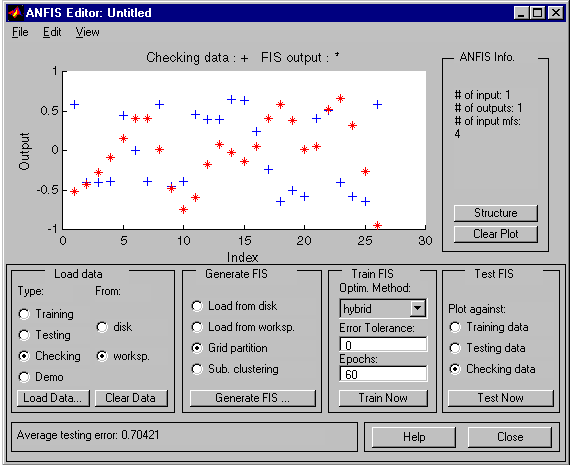
Proses adaptasi yang terjadi dalam sistem ANFIS dikenal juga dengan pembelajaran. Parameter-parameter ANFIS (baik premis maupun konsekwen) Selama proses belajar akan diperbaharui menggunakan metode pembelajaran. Metode pembelajaran yang digunakan dalam sistem ANFIS adalah algoritma pembelajaran hibrid. Algoritma ini terdiri dari dua bagian yaitu bagian arah maju dan bagian arah mundur. Pada bagian arah maju, proses adaptasi dilakukan menggunakan metode LSE dan terjadi pada parameter konsekwen. Sedangkan pada bagian arah mundur, proses adaptasi dilakukan menggunakan metode gradient-descent dan terjadi pada parameter premise.

Untuk mempelajari fuzzy dan ANFIS, paling mudah melalui MATLAB. Ketiklah fuzzy pada command prompt MATLAB maka akan tampil FIS editor, lalu bukalah file **tipper.fis** pada folder MATLAB sebagai demo Fuzzy inference system untuk pemberi tips pada pelayanan rumah makan:

****

10.5 FIS editor dengan tipper.fis

Untuk mencoba ANFIS pada MATLAB, ketiklah Anfisedit, akan tampil ANFIS editor. Lalu cobalah load data percobaan dan pelatihan seperti pada tutorial ANFIS editor di MATLAB, maka akan tampil hasil seperti berikut :

****

10.6 Hasil checking ANFIS Editor

Berikut contoh ANFIS pada inverse learning, terlihat bahwa sistem dapat memberikan hasil yang cukup memuaskan :

**plant.m:**

function out = plant(y, u);

out = y\*u/(1+y^2)-tan(u);

% Get data for inverse control

t = (0:100)';

u = 0.8\*sin(2\*pi\*t/250)+0.2\*sin(2\*pi\*t/50);

u = 2\*rand(size(t))-1;

y = ones(length(t), 1);

y(1) = 0.9;

% find the output y(k)

for i = 1:length(t)-1;

y(i+1) = plant(y(i), u(i));

end

genfig('Collecting data for inverse learning');

blackbg;

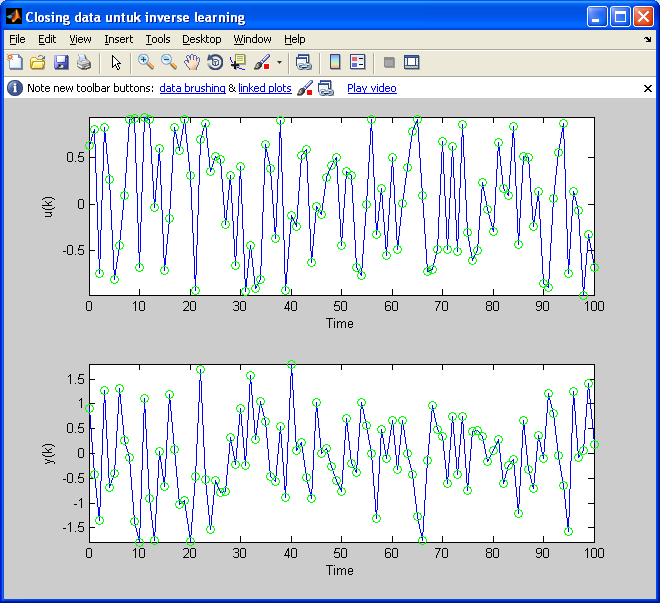
subplot(211); plot(t, u, '-', t, u, 'go');

xlabel('Time'); ylabel('u(k)'); axis([-inf inf -inf inf]);

subplot(212); plot(t, y, '-', t, y, 'go');

xlabel('Time'); ylabel('y(k)'); axis([-inf inf -inf inf]);

drawnow



Gambar 12.6 Data inverse learning ANFIS

% collect training data of the format: [y(k) y(k+1); u(k)]

% this is forward training data

col1 = y; col1(length(col1)) = [];

col2 = y; col2(1) = [];

col3 = u; col3(length(col3)) = [];

data = [col1 col2 col3];

trn\_data = data;

x\_data = trn\_data(:, 1);

y\_data = trn\_data(:, 2);

z\_data = trn\_data(:, 3);

% plotting training data as a scatter plot

genfig('training data distribution');

blackbg;

subplot(2,2,1); plot(x\_data, y\_data, 'o');

xlabel('y(k)'); ylabel('y(k+1)'); title('Training Data');

axis equal; axis square;

subplot(2,2,2); plot3(x\_data, y\_data, z\_data,'o');

axis([-inf inf -inf inf -inf inf]);

set(gca, 'box', 'on');

xlabel('y(k)'); ylabel('y(k+1)'); title('Training Data');

drawnow

% generate initial FIS matrix

mf\_n = 3;

epoch\_n = 50;

ss = 0.1;

mf\_type = 'gbellmf';

in\_fismat = genfis1(trn\_data, mf\_n, mf\_type);

% start training

[trn\_out\_fismat trn\_error step\_size] = ...

anfis(trn\_data, in\_fismat, [epoch\_n 0 ss], [1,1,1,1]);

% compute the result

z\_hat = evalfis([x\_data y\_data], trn\_out\_fismat);

genfig('Error curve and step sizes');

blackbg;

subplot(2,2,1); plot(trn\_error);

xlabel('Epochs'); ylabel('Root-Mean-Squared Error');

title('Error Curves');

subplot(2,2,2); plot(step\_size);

xlabel('Epochs'); title('Step Sizes');

drawnow

genfig('Initial and final MFs');

blackbg;

% plot initial MF's on x and y

subplot(2,2,1);

[mfx, mfy] = plotmf(in\_fismat, 'input', 1);

plot(mfx, mfy); title('(a) Initial MFs on y(k)'); axis([-inf inf 0 1.2]);

subplot(2,2,3);

[mfx, mfy] = plotmf(in\_fismat, 'input', 2);

plot(mfx, mfy); title('(a) Initial MFs on y(k+1)'); axis([-inf inf 0 1.2]);

% plot final MF's on x and y

subplot(2,2,2);

[mfx, mfy] = plotmf(trn\_out\_fismat, 'input', 1);

plot(mfx, mfy); title('(a) Final MFs on y(k)'); axis([-inf inf 0 1.2]);

subplot(2,2,4);

input\_index = 2;

[mfx, mfy] = plotmf(trn\_out\_fismat, 'input', 2);

plot(mfx, mfy); title('(a) Final MFs on y(k+1)'); axis([-inf inf 0 1.2]);

drawnow

% ANFIS surface plot

genfig('ANFIS surface');

blackbg;

point\_n = 20;

x\_tmp = linspace(min(x\_data), max(x\_data), point\_n);

y\_tmp = linspace(min(y\_data), max(y\_data), point\_n);

[xx, yy] = meshgrid(x\_tmp, y\_tmp);

zz = evalfis([xx(:) yy(:)], trn\_out\_fismat);

subplot(2,2,1);

mesh(xx, yy, reshape(zz, point\_n, point\_n));

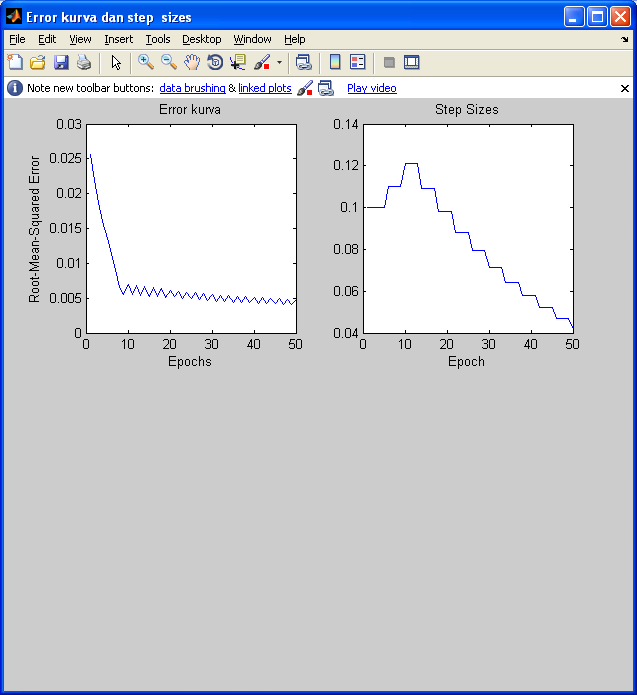
hold on; plot3(x\_data, y\_data, z\_data, 'o'); hold off;

axis([-inf inf -inf inf -inf inf]);

set(gca, 'box', 'on');

xlabel('y(k)'); ylabel('y(k+1)'); title('ANFIS Output');

view([240 60]);



Gambar 10.7 Error kurva dan step sizes

% verifikasi kontroler yang diperoleh

% Verify the obtained controller

t = (0:300)';

u = 2\*rand(size(t))-1;

desired\_y = 0.6\*sin(2\*pi\*t/250)+0.2\*sin(2\*pi\*t/50);

%desired\_y = 0.8\*cos(2\*pi\*3\*t/100);

y = ones(length(t), 1);

y(1) = 0.0; % this is not important

for i = 1:length(t)-1;

% Assuming we DO know desired\_y(i+1);

u(i) = evalfis([y(i) desired\_y(i+1)], trn\_out\_fismat);

y(i+1) = plant(y(i), u(i));

end

genfig('Application of inverse fuzzy controller');

blackbg;

subplot(2,2,1); plot(t, desired\_y, ':', t, y, '-');

xlabel('Time'); ylabel('Actual and Desired y(k)');

legend('desired', 'actual');

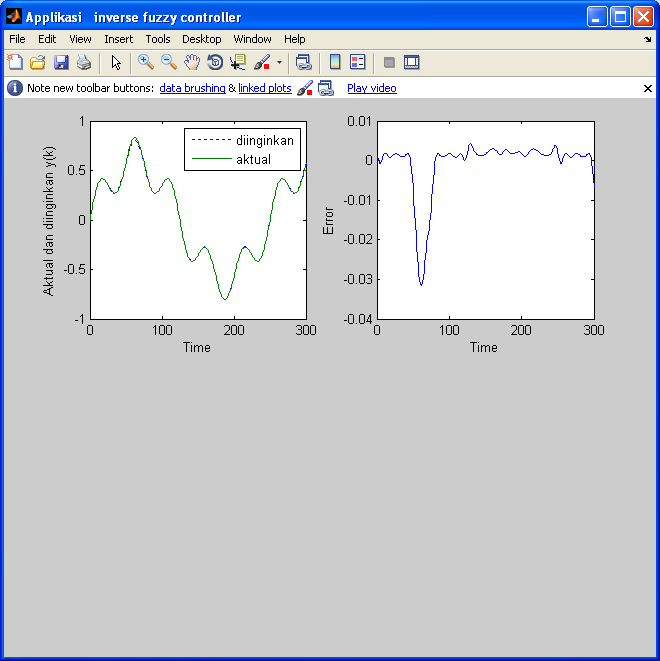
subplot(2,2,2); plot(t, desired\_y-y);

xlabel('Time'); ylabel('Error');

%subplot(2,2,3); plot(t, u);

%xlabel('Time'); ylabel('u(k)');

%surfview(trn\_out\_fismat);

****

Gambar 10.8 Hasil kontroler ANFIS

**SIMPULAN**

*Pada dasarnya, teknik Soft Computing terdiri dari Fuzzy logic, neuro computing atau Artificial neural network (ANN), Evolutionary Computation dan probabilistic computing. ANFIS ini merupakan gambungan dari kehebatan logika fuzzy dan neural networks. Logika fuzzy juga dapat diterapkan pada sistem kontrol pada robot. Logika Fuzzy (logika samar) merupakan logika yang berhadapan langsung dengan konsep kebenaran sebagian, dimana logika klasik menyatakan bahwa segala hal dapat diekspresikan  dalam binary 0 atau 1.*

**DAFTAR PUSTAKA**

* Adries P. Engelbrect. (2007), ***Computational Intelligence An Introduction***. 2nd ed. John Wiley & Sons. USA.
* Zadeh, Lotfi A. (1997), “**What is Soft Computing**”, Springer Verlag Germany.
* J .S. Jang (1997), **Neuro Fuzzy And Soft Computing**, Prentice Hall Publisher, Upper Saddle River, NJ.
* http://www.mathworks.com/products/fuzzy-logic/