

MODUL PRAKTIKUM

KOMPUTASI CERDAS

NAMA MAHASISWA

NIM MAHASISWA

lABORATORIUM TEKNIK ELEKTRO

jURUSAN tEknik ELEKTRO

FAKULTAS TEKNik

universitas muhammadiyah malang

2020

DAFTAR ISI

[DAFTAR ISI iii](#_Toc530907266)

[DAFTAR GAMBAR iv](#_Toc530907267)

[DAFTAR TABEL v](#_Toc530907268)

[BAB 1 OPTIMASI MENGGUNAKAN PSO 1](#_Toc530907269)

[1.1 Latar Belakang 1](#_Toc530907270)

[1.2 Tujuan 1](#_Toc530907271)

[1.3 Dasar Teori 1](#_Toc530907272)

[1.3.1 Algoritma PSO 2](#_Toc530907273)

[1.3.2 Contoh Implementasi PSO 2](#_Toc530907274)

[1.3.3 Implementasi PSO dengan Matlab: 5](#_Toc530907275)

[BAB 2 Artificial Neural Network 7](#_Toc530907276)

[2.1 Tujuan 7](#_Toc530907277)

[2.2 Dasar Teori 7](#_Toc530907278)

[2.2.1 Jenis Jenis hidden layer 8](#_Toc530907279)

[2.2.2 Multilayer Neural Network (Backpropagation) 1](#_Toc530907280)

[2.2.3 Pelatihan bobot pada multilayer neural network 2](#_Toc530907281)

[2.2.4 Backpropagation Pada Permasalahan Prakiraan Cuaca dengan Matlab 3](#_Toc530907282)

[2.2.5 Langkah Langkah Percobaan 4](#_Toc530907283)

[BAB 3 FUZZY LOGIC 7](#_Toc530907284)

[3.1 Tujuan 7](#_Toc530907285)

[3.2 Dasar Teori 7](#_Toc530907286)

[3.2.1 Fuzzification 8](#_Toc530907287)

[3.2.2 Rule Base 10](#_Toc530907288)

[3.2.3 Inference Mechanism (Operasi Himpunan) 10](#_Toc530907289)

[3.2.4 Defuzzification 11](#_Toc530907290)

[3.3 Fuzzy Logic Control 11](#_Toc530907291)

[3.4 Pemrograman Fuzzy pada contoh 3.3 14](#_Toc530907292)

[DAFTAR PUSTAKA](#_Toc530907293)

DAFTAR GAMBAR

[Gambar 2.1 neuron pada jaringan syaraf biologis 7](#_Toc530906688)

[Gambar 2.2 Model neuron pada ANN 8](#_Toc530906689)

[Gambar 2.3 grafik nilai keluaran fungsi aktifasi sigmoid 1](#_Toc530906690)

[Gambar 2.4 Struktur Multilayer Neural Network 1](#_Toc530906691)

[Gambar 3.1 Struktur Sistem Fuzzy 7](#_Toc530906692)

[Gambar 3.2 Perbandingan keanggotaan bolean dan fuzzy 8](#_Toc530906693)

[Gambar 3.3 grafik fungsi keanggotaan segitiga 9](#_Toc530906694)

[Gambar 3.4 block diagram Fuzzy logic control 12](#_Toc530906695)

[Gambar 3.5 membership function error 12](#_Toc530906696)

[Gambar 3.6 membership function delta error 12](#_Toc530906697)

[Gambar 3.7 membership function sinyal control 12](#_Toc530906698)

[Gambar 3.8 blok Simulink fuzzy logic control pada kecepatan motor 16](#_Toc530906699)

DAFTAR TABEL

[Tabel 1.1 Hasil Percobaan 1.1 6](#_Toc530906734)

[Tabel 1.2 Hasil Percobaan 1.2 6](#_Toc530906735)

[Tabel 2.1 Tabel kriteria cuaca didaerah X [] 3](#_Toc530906736)

[Tabel 2.2 Ketentuan data input unsur cuaca dengan input neuron 3](#_Toc530906737)

[Tabel 2.3 Data output hasil modifikasi 4](#_Toc530906738)

[Tabel 2.4 Data Training 4](#_Toc530906739)

[Tabel 2.5 Data Hasil Percobaan 1 6](#_Toc530906740)

[Tabel 3.1 Derajat keanggotaan bolean dan fuzzy 8](#_Toc530906741)

[Tabel 3.2 contoh tabel basis aturan fuzzy 10](#_Toc530906742)

[Tabel 3.3 Tabel basis aturan fuzzy logic control 12](#_Toc530906743)

# OPTIMASI MENGGUNAKAN PSO

## Latar Belakang

Pengertian optimasi secara umum yaitu proses pencarian satu atau beberapa variable agar tercapai suatu tujuan optimal (maks/min). Permasalahan optimasi merupakan permasalahan yang sering kita hadapi khususnya dalam bidang teknik. Contoh contoh permasalahan optimasi yang umum sering ditemui seperti pencarian jalur terdekat, dan penjadwalan. Sedangkan contoh-contoh permasalahan optimasi pada bidang Teknik seperti pembangkitan daya generator, penjadwalan daya energi terbaharukan (hybrid), dan pencarian parameter pada sebuah system.

Terdapat beberapa metode penyelesaian optimasi seperti penyelesaian secara matematis (gradien), metode klasik (numerik) seperti Newton Raphson, dan penyelesaian dengan metode heuristic atau metode cerdas. Metode heuristik saaat ini telah banyak digunakan untuk menyelesaiakan permaslahan optimisasi dikarenakan penyelesaian secara matematis tidak mungkin dilakukan dikarenakan keterbatasan dalam proses turunan, kemudian keterbatasan pada metode numerik yang hanya dapat menyelesaikan permasalahan linear.

## Tujuan

Mahasiswa dapat mengimplementasikan PSO sebagai metode penyelesaian permasalahan optimisasi.

## Dasar Teori

PSO atau particle swarm optimization merupakan salah satu algoritma cerdas atau metode heuristik yang terinspirasi oleh burung atau ikan yang sedang bermigrasi. Proses pencarian variable dalam PSO diibaratkan dengan sekumpulan burung atau ikan yang dalam pso disebut sebuah particle yang terbang atau berenang menyusuri area atau ruang yang selanjutnya diibaratkan dengan ruang permasalahan (fungsi tujuan). Saat sekumpulan burung terbang atau sekumpulan ikan sedang berenang memiliki beberapa sifat berikut:

1. Meskipun berpindah secara bersama tetapi tidak saling bertabrakan atau bersinggungan (sparasi).
2. Pencarian Bersama (Kohesi)
3. Penyesuaian (Aligment)

Pada PSO particle berpindah dengan menggunakan persamaan berikut:

|  |  |
| --- | --- |
|  | (1) |

dengan perpindahan posisi partikel menggunakan persamaan berikut:

|  |  |
| --- | --- |
|  | (2) |

### Algoritma PSO

Algoritma PSO dijabarkan dalam listing program berikut:

|  |
| --- |
| *Start**Inisialisasi posisi particle awal sejumlah N**Inisialisasi kecepatan particle awal* *Inisialisasi parameter dan* *Evaluasi terhadap fungsi tujuan* *Cari Particle terbaik dari kumpulan particle**While (Iterasi < Max Iterasi)* *For i=1:N* *For d=1:D* *Hitung kecepatan dan posisi partikel dengan pers* (1) *dan pers* (2) *End*  *End* *Evaluasi terhadap fungsi tujuan* *Cari indeks pengalaman terbaik partikel* *Cari indeks partikel terbaik* *End* |
|  |

### Contoh Implementasi PSO

Misal kita mempunyai sebuah permasalahan optimasi dengan fungsi tujuan berikut:

dimana

Iterasi 0:

1. Tentukan jumlah populasi particle N=4
2. Tentukan populasi awal secara random sesuai dengan batas atas dan batas bawah
3. Tentukan Kecepatan awal
4. Tentukan
5. Evaluasi terhadap
6. Tentukan
7. Dari evaluasi no 4 cari

Dikarenakan pada contoh ini permasalahan merupakan minimasi maka particle terbaik adalah yang menghasilkan nilai fungsi tujuan terkecil

Iterasi 1:

1. Hitung kecepatan dan posisi partikel. Missal nilai random yang didapat adalah

sehingga posisi partikel adalah:

1. Evaluasi terhadap
2. Tentukan

Pada persoalan minimasi untuk mencari maka dilakukan perbandingan antara dengan apakah lebih kecil dibanding dengan ketentuan sebagai berikut:

1. Dari evaluasi no 3 cari

Iterasi 2:

1. Hitung kecepatan dan posisi partikel. Missal nilai random yang didapat adalah

sehingga posisi partikel adalah:

1. Evaluasi terhadap
2. Tentukan
3. Dari evaluasi no 3 cari

### Implementasi PSO dengan Matlab:

Pada subab ini PSO dimplementasikan untuk mengoptimalkan sebuah fungsi tujuan (*Fitness*) agar fungsi tersebut bernilai minimum.

 dimana

Dengan

Langkah Langkah Percobaan:

1. Membuat source code fungsi tujuan pada matlab sebagai berikut:

|  |
| --- |
| function f=fitness(x)a=3\*(x(1,1))^2+4\*(x(1,2))^2+2\*(x(1,3))^2;b=-3\*x(1,1)\*x(1,2)-2\*x(1,1)\*x(1,3)-3\*x(1,2)\*x(1,3);c=-5\*x(1,1)-4\*x(1,2)-6\*x(1,3);f=a+b+c;end |

1. Simpan dengan nama fitness.m
2. Membuat source code program utama PSO pada matlab sebagai berikut:

|  |
| --- |
| %%Inisialisasi PSO%%jumlah\_particle=20;%%20-30 particleD=3;batas\_atas=15;batas\_bawah=0;x=(batas\_atas-batas\_bawah).\*rand(jumlah\_particle,D)... +repmat(batas\_bawah,jumlah\_particle,D)v=zeros(jumlah\_particle,D)P=zeros(jumlah\_particle,D) f=zeros(jumlah\_particle,1)Max\_Iter=10000c1=1;%%0-2c2=1;%%0-2minftot=[];%%index Nilai fungsi paling minimum%%Iterasi Ke-0%%for i=1:jumlah\_particle f(i,:)=fitness(x(i,:));endP=x;fbest=f;[minf,idk]=min(f);G=x(idk,:); for t=1:Max\_Iter for d=1:D for i=1:jumlah \_particle v(i,d)=v(i,d)+(c1\*rand)\*(P(i,d)-x(i,d))... +(c2\*rand)\*(G(:,d)-x(i,d)); x(i,d)=v(i,d)+x(i,d); end end for i=1:jumlah\_particle f(i,:)=fitness(x(i,:)); end changerow=f<fbest; fbest=fbest.\*(1-changerow)+f.\*changerow; P(changerow,:)=x(changerow,:);  [minf,idk]=min(fbest); minftot=[minftot;minf]; G=P(idk,:);endx\_optimal=Gminimum\_f=minfplot(minftot) |

1. Simpan dengan nama PSO.m
2. Run program PSO.m
3. Catat hasil pada table berikut:

Tabel 1.1 Hasil Percobaan 1.1

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

1. Pada source code program utama PSO ubah persamaan kecepatan menjadi:

|  |
| --- |
| v(i,d)=0.75\*v(i,d)+(c1\*rand)\*(P(i,d)-x(i,d))... +(c2\*rand)\*(G(:,d)-x(i,d)); |

1. Run kembali program PSO.m
2. Catat hasil pada table berikut:

Tabel 1.2 Hasil Percobaan 1.2

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

1. Analisa hasil percobaan pada Tabel 1.2 dengan menggunakan metode matematis untuk menentukan nilai *x1, x2, x3, f(x)*

## Data Hasil Percobaan

Tabel 1.1 Hasil Percobaan 1.1

Tabel 1.2 Hasil Percobaan 1.2

## Analisa Perhitungan

## Analisa Data

**1.7 Kesimpulan**

# Artificial Neural Network

## Tujuan

Mahasiswa dapat mengimplementasikan ANN sebagai metode penyelesaian permasalahan identifikasi, estimasi, dan prediksi dari ebuah system.

## Dasar Teori

Artificial Neural Network (ANN) atau sering disebut juga dengan jaringan syaraf tiruan adalah system pemroses informasi yang memiliki karakteristik menyerupai dengan karakteristik jaringan syaraf biologi.

ANN dibentuk dengan generalisasi model dari jaringan syaraf biologi dengan asumsi sebagai berikut:

* Pemrosesan informasi terjadi pada banyak elemen sederhana (neuron)
* Sinyal dikirimkan diantara neuron neuron melalui penghubung penghubung
* Penghubung antar neuron memiliki bobot yang akan memperkuat atau memperlemah sinyal

Pada neuron jaringan syaraf biologis terdapat tiga komponen penyusun utama yaitu *Dendrit*, *Soma*, dan *Axon*. *Dendrit* berperan sebagai bagian penerima informasi yang bisa saja berasal dari neuron yang lain. *Soma* berfungsi sebagai penampung atau pengolah sinyal, sedangkan *Axon* berfungsi sebagai penerus signal. Struktur neuron pada jaringan syaraf biologis ditunjukan pada Gambar 2.1.



Gambar 2.1 neuron pada jaringan syaraf biologis

Struktur neuron pada jaringan biologis tersebut kemudian digeneralkan dalam sebuah model matematika dengan asumsi asumsi yang sudah dijelaskan dijabarkan pada persamaan (3) dan (4):

|  |  |
| --- | --- |
|  | (3) |
|  | (4) |

Dimana:

 : input neuron

 : bobot neuron

 : bias

 : fungsi aktifasi

 : output neuron

 : jumlah input

Model matematika pada persamaan (3) dan (4) jika digambarkan memiliki struktur yang menyerupai jaringan syaraf biologis, seperti yang ditunjukan pada Gambar 2.2



Gambar 2.2 Model neuron pada ANN

### Jenis Jenis hidden layer

1. Fungsi Aktifasi Threshold

Persamaan fungsi aktifasi:

1. Fungsi Aktifasi Linear

Persamaan fungsi aktifasi:

Turunan fungsi aktifasi:

1. Fungsi Aktifasi Sigmoid

Persamaan fungsi aktifasi:



Gambar 2.3 grafik nilai keluaran fungsi aktifasi sigmoid

Turunan fungsi aktifasi:

### Multilayer Neural Network (Backpropagation)

Multilayer neural network merupakan ANN yang terdapat banyak neuron pada strukturnya. Neuron tersebut disusun secara umum menjadi 3 bagian layer yaitu layer *input*, layer *hidden*, dan layer *output* yang digambarkan pada Gambar 2.4.



Gambar 2.4 Struktur Multilayer Neural Network

Multilayer neural network memiliki model matematis yang dijabarkan pada persamaan (5) hingga persamaan (8). Dimana persamaan *forward* dari input ke *hidden layer* dijabarkan pada persamaan (5) dan (6)

|  |  |
| --- | --- |
|  | (5) |
|  | (6) |

Kemudian persamaan forward dari *hidden layer* ke *output layer* dijabarkan pada persamaan (7) dan (8)

|  |  |
| --- | --- |
|  | (7) |
|  | (8) |

Dengan

|  |  |  |
| --- | --- | --- |
|  | : | Bobot input ke *hidden layer* pada input i dan *hidden layer* ke j |
|  | : | Nilai input ke i dan pada data ke l |
|  | : | Bobot bias input ke *hidden layer* ke j |
|  | : | Output pada *hidden layer* ke j  |
|  | : | Bobot *hidden* ke *output layer* pada *hidden* ke j dan *output* ke k |
|  | : | Bobot bias input ke *output layer* ke k |
|  | : | Output ke k |
|  | : | Fungsi aktifasi pada *hidden layer* ke j |
|  | : | Fungsi aktifasi pada *output layer* ke k |

### Pelatihan bobot pada multilayer neural network

Pelatihan bobot pada multilayer neural network menggunakan metode *least square* dimana bobot pada *hidden* ke *output layer* dilatih terlebih dahulu berdasarkan error yang terjadi pada perbandingan nilai antara output layer dengan output pada data target atau data *training.*

Setelah proses pelatihan bobot pada *hidden* ke *output layer* dilakukan maka kemudian proses pelatihan bobot pada input ke *hidden layer* dilakukan dengan asumsi bahwa error yang terjadi pada output layer merambat *(propagate)* secara merata pada hidden layer.

Persamaan perubahan bobot pada *hidden* ke *output layer* dijabarkan pada persamaan (9) dan (10):

|  |  |
| --- | --- |
|  | (9) |
|  | (10) |

dengan persamaan perubahan bias pada *output layer* dijabarkan pada persamaan (11) dan (12):

|  |  |
| --- | --- |
|  | (11) |
|  | (12) |

Sedangkan persamaan perubahan bobot pada *input* ke *hidden* dijabarkan pada persamaan (13) dan (14):

|  |  |
| --- | --- |
|  | (13) |
|  | (14) |

dengan persamaan perubahan bias pada *hidden layer* dijabarkan pada persamaan(15) dan (16):

|  |  |
| --- | --- |
|  | (15) |
|  | (16) |

dimana

|  |  |  |
| --- | --- | --- |
|  | : | Perubahan Bobot pada input ke i dan *hidden layer* ke j |
|  | : | Perubahan Bobot *hidden* ke j dan *output* ke k |
|  | : | Perubahan Bobot bias pada *hidden layer* ke j |
|  | : |  |
|  | : | Bobot bias input pada *output layer* ke k |
|  | : | Learning rate |
|  | : | Turunan Fungsi aktifasi pada *hidden layer* ke j |
|  | : | Turunan Fungsi aktifasi pada *output layer* ke k |

### Backpropagation Pada Permasalahan Prakiraan Cuaca dengan Matlab

Pada Implementasi ini kita akan memodelkan system parakiraan cuaca secara sederhana. Dari data suhu, arah angin, kelembaban, dan tekanan udara diklasifikasikan pada Tabel 2.1

Tabel 2.1 Tabel kriteria cuaca didaerah X []

|  |  |
| --- | --- |
| **Unsur Cuaca** | **Keadaan Cuaca** |
| **Cerah** | **Berawan** | **Hujan** | **Hujan Lebat** |
| **Arah Angin (degree)** | <150 | 150-200 | >200 | >200 |
| **Suhu (Derajat Celcius)** | >29 | 26-29 | 26-29 | <26 |
| **Kelembaban(%)** | <70 | 70-85 | >80 | >80 |
| **Tekanan Udara** | >1010 | 1007-1010 | 1007-1010 | <1007 |

Pada Tabel 2.1 input merupakan unsur cuaca dan output yang dihasilkan adalah keadaan cuaca. Data input meliputi arah angin, suhu, kelembaban, dan tekanan udara. Sehingga input pada neural network berjumlah 4 buah dan output 1 buah yaitu keadaan cuaca. Agar proses learning lebih mudah untuk membuat system prakiraan cuaca dengan Neural Network berdasarkan karakteristik tersebut terlebih dahulu data dimodifikasi dengan ketentuan sebagai berikut:

Tabel 2.2 Ketentuan data input unsur cuaca dengan input neuron

|  |  |
| --- | --- |
| Unsur Cuaca | Rentang Variable |
| Arah Angin | <150 | 150-200 | >200 |
| X1 | -1 | 0 | 1 |
| Suhu | <26 | 26-29 | >29 |
| X2 | -1 | 0 | 1 |
| Kelembaban | <70% | 70%-85% | >85% |
| X3 | -1 | 0 | 1 |
| Tekanan | <1007  | 1007-1010 | >1010 |
| X4 | -1 | 0 | 1 |

Untuk data output yaitu keadaan cuaca terdapat 4 kondisi dimana pada NN fungsi aktifasi memiliki rentang output 0-1 sehingga output neuron memiliki retang nilai ¼ sehingga data output sebagai berikut:

Tabel 2.3 Data output hasil modifikasi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Keadaan Cuaca | Cerah | Berawan | Hujan | Hujan Lebat |
| yt | 0.25 | 0.5 | 0.75 | 1 |

Sehingga data training keseluruhan disajikan pada

Tabel 2.4 Data Training

|  |  |
| --- | --- |
| **Input Neuron** | **Output**  |
| Arah Angin (X1) | suhu (X2) | Kelembaban (X3) | Tekanan Udara (X4) |
| -1 | 1 | -1 | 1 | 0.25 |
| 0 | 0 | 0 | 0 | 0.5 |
| 1 | 0 | 1 | 0 | 0.75 |
| 1 | -1 | 1 | -1 | 1 |

## Langkah Langkah Percobaan

1. Buat program pada matlab dengan listing sebagai berikut

*Listing program ANN untuk permasalahan Prakiraan Cuaca*

|  |
| --- |
| *clear**clc**%==================Inisialisali Parameter NN=================**jmlh\_input=4;**jmlh\_hidden=15;**jmlh\_output=1;**wih=zeros(jmlh\_input,jmlh\_hidden);**woh=zeros(jmlh\_hidden,jmlh\_output);**bih=zeros(1,jmlh\_hidden);**boh=zeros(1,jmlh\_output);**max\_iter=10000;**lamdha=0.4;**jmlh\_data=4;**%==================Inisialisali Parameter NN=================**x=[-1 1 -1 1;* *0 0 0 0;* *1 0 1 0;* *1 -1 1 -1;];**yt=[0.25; %cerah 0-0.25* *0.5; %berawan 0.25-0.5* *0.75; %Hujan 0.5 - 0.75* *1;]; %Hujan Lebat0.75-1**max\_error=[];**for t=1:max\_iter* *for l=1:jmlh\_data* *for j=1:jmlh\_hidden* *zih(j)=0;* *for i=1:jmlh\_input* *zih(j)=zih(j)+wih(i,j)\*x(l,i);* *end* *yih(j)=1/(1+exp(-(zih(j)+bih(j))));* *end* *for k=1:jmlh\_output* *zoh(k)=0;* *for j=1:jmlh\_hidden* *zoh(k)=zoh(k)+woh(j,k)\*yih(1,j);* *end* *yoh(k)=1/(1+exp(-(zoh(k)+boh(k))));* *end* *%Perhitungan Backward* *for k=1:jmlh\_output* *error(k)=yt(l,k)-yoh(k)* *dboh(1,k)=lamdha\*(error(k))\*(1-yoh(k))\*yoh(k);* *boh(1,k)=boh(1,k)+dboh(1,k); %Menghitung Bias baru* *for j=1:jmlh\_hidden* *dwoh(j,k)=lamdha\*(error(k))\*(1-yoh(k))\*yoh(k)\*yih(j);%% Perubahan bobot* *woh(j,k)=woh(j,k)+dwoh(j,k);%% Menghitung Bobot baru* *errh(j,k)=error(k)/jmlh\_hidden;* *end* *for j=1:jmlh\_hidden* *bih(j)=bih(j)+lamdha\*errh(j,k)\*(1-yih(j))\*yih(k);* *for i=1:jmlh\_input* *dwih(i,j)=lamdha\*errh(j,k)\*(1-yih(j))\*yih(j)\*x(l,i);* *wih(i,j)=wih(i,j)+dwih(i,j);* *end* *end* *end* *end**max\_error=[max\_error,max(error)];%% Mengindexkan nilai error tiap iterasi**end**plot(max\_error)* |

1. Simpan Listing program dengan nama *backpropagation*

**

1. Buka tab baru untuk membuat listing baru
2. Buat program pada matlab dengan listing sebagai berikut

*Listing program ANN untuk test data*

|  |
| --- |
| function y=coba\_forward(x,woh,boh,wih,bih,jmlh\_input,jmlh\_hidden,jmlh\_output) for j=1:jmlh\_hidden zih(j)=0; for i=1:jmlh\_input zih(j)=zih(j)+wih(i,j)\*x(1,i); end yih(j)=1/(1+exp(-(zih(j)+bih(j)))); end for k=1:jmlh\_output zoh(k)=0; for j=1:jmlh\_hidden zoh(k)=zoh(k)+woh(j,k)\*yih(1,j); end yoh(k)=1/(1+exp(-(zoh(k)+boh(k)))); y(k)=yoh(k); endend |

1. *Run* listing program backpropagation dan simpan data grafik error pada hasil percobaan
2. Uji data output pada neural network dengan data training dengan mengetikan listing berikut pada *Command Window*



1. *Tekan Enter dan Catat hasil keluaran neural network pada* Tabel 2.5

Tabel 2.5 Data Hasil Percobaan 1

|  |  |
| --- | --- |
| **Input Neuron** | **y** |
| Arah Angin (X1) | suhu (X2) | Kelembaban (X3) | Tekanan Udara (X4) |
| -1 | 1 | -1 | 1 | 0.2378 |
| 0 | 0 | 0 | 0 | 0.5082 |
| 1 | 0 | 1 | 0 | 0.7575 |
| 1 | -1 | 1 | -1 | 0.9640 |

1. Ulangi langkah 6-7 untuk data masukan lain sehingga Tabel 2.5 terpenuhi
2. Jika **y** bernilai ***0-0.25*** adalah nilai **cerah(C) ,** **berawan(B)** **y** bernilai *0.25-0.5,* **Hujan(H)** **y** bernilai *0.5 - 0.75*, **Hujan Lebat(HL)** bernilai *lebih dari 0,75* ulangi langkah 6-7 untuk data input berikut

Tabel 2.6 Data Hasil Percobaan 2

|  |  |  |
| --- | --- | --- |
| **Input Neuron** | **y** | **C/B/H/HL** |
| Arah Angin (X1) | suhu (X2) | Kelembaban (X3) | Tekanan Udara (X4) |
| 0 | 1 | -1 | 1 | 0.2499 | C |
| 1 | 0 | 0 | 0 | 0.5025 | H |
| 0 | 0 | 1 | 0 | 0.7549 | HL |
| 1 | 1 | 1 | 0 | 0.9669 | HL |

1. Ubah nilai *lamdha* menjadi 0.6 dan run kembali listing backpropagation, Simpan dan amati data grafik error.

## Data Hasil Percobaan

Tabel 2.5 Data Hasil Percobaan 1

Tabel 2.6 Data Hasil Percobaan 2

Grafik 2.1 Lamdha 0.4

Grafik 2.1 Lamdha 0.6

## Analisa Data

**2.6 Kesimpulan**

#  FUZZY LOGIC

## Tujuan

Mahasiswa dapat mengimplementasikan Fuzzy sebagai metode penyelesaian permasalahan penggolongan data.

## Dasar Teori

Fuzzy Logic merupakan peningkatan dari logika bolean, dimana pada logika bolean keanggotaan hanya dinyatakan dengan “iya” dan “tidak”. Fuzzy Logic salah satu pendekatan dimana representasi suatu kejadian didistribusikan kedalam sejumlah istilah bahasa (yang menyatakan level kualitatif).Fuzzy logic memiliki pendekatan kepada intuisi manusia yang menyatakan sesuatu dengan tingkatan seperti menyatakan kondisi suhu dengan tingkatan “sangat dingin”, “dingin”, “sedang”, “tidak dingin”, “panas”. Sistem fuzzy logic digambarakan dalam diagram pada Gambar 3.1, dimana pada digram tersebut terdapat 4 bagian utama yaitu 1). Fuzzification 2). Inference Mechanism 3). Rule-Base 4). Defuzzification.



Gambar 3.1 Struktur Sistem Fuzzy

### Fuzzification

Fuzzifikasi merupakan proses penggolongan atau perubahan nilai pada variable input kedalam fuzzy set (himpunan fuzzy). Input pada fuzzy bisa terdiri dari banyak variable, dimana pada masing masing variable akan digolongkan pada masing masing himpunan fuzzy. Contoh 3.1:

Masukan fuzzy berupa variable suhu dan kecepatan motor, dengan himpunan variable input tersebut adalah sebagai berikut:

1. **Fuzzy Set**

Fuzzy set atau himpunan fuzzy adalah suatu himpunan yang beranggotakan sejumlah istilah dalam pengertian bahasa yang menyatakan level kualitatatif dari semesta pembicaraan X, seperti pada contoh 3.1 semesta suhu digolongkan menjadi 5 tingkatan himpunan. Keanggotaan dalam fuzzi set dengan logika manusia sangat kompleks, sehingga tidak dapat dinyatakan dalam bentuk tertentu dan berbeda untuk tiap individu.

1. **Fuzzy Membership function**

Fungsi keanggotaan fuzzy adalah suatu fungsi yang didefinisikan untuk suatu anggota himpunan fuzzy yang menggambarkan derajat kebenaran suatu kejadian dalam semesta pembicaan X, dinyatakan dalam tingkat keanggotaan (grade of membership) dengan nilai antara 0 s/d 1. Untuk menggambarkan bagaimana fingsi keanggotaan fuzzy perhatikan contoh 3.2:

Perbandingan Nilai Kebenaran antara Logika Bolean dengan Logika Fuzzy dalam kasus kejadian tertentu.

X: Semesta Pembicaraan “Kecepatan Putaran Motor (rpm)”

Kejadian x: kecepatan 1200 rpm

|  |  |
| --- | --- |
| fuzzy vs booleanGambar 3.2 Perbandingan keanggotaan bolean dan fuzzy | Tabel 3.1 Derajat keanggotaan bolean dan fuzzy |

Pada Gambar 3.2 dan Tabel 3.1 menggambarkan perbandingan keanggotaan dari logika bolean dan fuzzy dimana pada kejadian kecepatan motor berputar 1200 nilai derajat keanggotaan sedang pada bolean bernilai 1 sedangkan pada fuzzy dinyatakan dalam tingkatan derajat keanggotaan yang berbeda. Pada kondisi tersebut nilai derajat keanggotan 0.8 pada himpunan sedang, 0.2 pada keanggotan lambat dan 0.6 pada keanggotan lambat.

1. **Fuzzy Membership function Representation**

Dalam bentuk representasi umum Himpunan dinyatakan dalam bentuk sebagai berikut:

S :{himpunan fuzzy semesta pembicaraan}

Si E S, i=1,2 ..n ; Si : himpunan pendukung ke i, dan n jumlah himpunan pendukung.

Nilai Logika Fuzzy X dapat dinyatakan dalam representasi umum himpunan:

X={µx1/S1; µx1/S1 ....... ; µxn/Sn}

Pada contoh 3.2 Kecepatan Putaran Motor dinyatakan dalam 5 keanggotaan

Dimana pada kondisi 1200 nilai keanggotaanya adalah sebagai berikut:

1. **Bentuk-Bentu Fungsi Keanggotaan**

Fungsi keanggotaan fuzzy memiliki bentuk yang beragam. Meskipun beragam fungsi keanggotaan tersebut tetap menyatakan derajat keanggotaan pada nilai 0 s/d 1. Bentuk bentuk fungsi keanggotaan tersebut diantaranya:

1. Distribusi-s
2. Distribusi-Z
3. Distribusi-Pi
4. Distribusi Trapesium
5. Distribusi Segitiga
6. Distribusi Gaussian

Fungsi keanggotaan dengan distribusi segitiga adalah yang paling umum digunakan karena kemudahanya. Fuzzifikasi dengan fungsi keanggotaan dengan distribusi segitiga dilakukan dengan persamaan :



Gambar 3.3 grafik fungsi keanggotaan segitiga

|  |  |
| --- | --- |
|  | (17) |

Contoh 3.3:

Berikut perhitungan pada fungsi keanggotaan LB, SD, dan CP pada contoh 3.2, dimana kondisi x=1200

### Rule Base

Basis Aturan fuzzy menyatakan hubungan kejadian yang ada pada input fuzzy dengan keputusan apa yang ada pada output fuzzy. Hubungan tersebut dapat dinyatakan dengan hubungan “jika” “maka” atau “if” “then”. Contoh 3.4, sebuah system control fuzzy dengan input fuzzy berupa nilai error ( dan nilai delta error ( dan output berupa sinyal control

jika istilah istilah dalam nilai error dan delta error

Dengan himpunan output sinyal control adalah sebagai berikut:

Maka basis aturan dari hubungan input dan output dapat dinyatakan dengan fungsi berikut:

|  |  |
| --- | --- |
|  | (18) |
|  | (19) |

Dan seterusnya hingga semua kondisi terpenuhi.

Selain disajikan dengan logika “if” “then”, basis aturan juga dapat disajikan dengan table sebagai berikut:

Tabel 3.2 contoh tabel basis aturan fuzzy

|  |  |
| --- | --- |
|  | Error |
| Delta error |  | nb | n | z | p | pb |
| nb | NB | NB | NB | N | Z |
| n | NB | NB | N | Z | P |
| z | NB | N | Z | P | PB |
| p | N | Z | P | PB | PB |
| pb | Z | P | PB | PB | PB |

### Inference Mechanism (Operasi Himpunan)

Infrensi fuzzy dilakukan untuk menghitung berapa nilai keanggotaan output berdasarkan nilai keanggotaan input dan basis aturan yang didefinisikan. Terdapat beberapa metode infrensi fuzzy diantaranya sebagai berikut:

1. Metode Generalize Modul Ponens (GMP) atau metode Mamdani:

|  |  |
| --- | --- |
|  | (20) |

1. Metode Larsent

|  |  |
| --- | --- |
|  | (21) |

### Defuzzification

Defuzzifikasi adalah bagian terakhir dari system fuzzy yang digunakan untuk menghitung besar nilai nyata berdasarkan hasil perhitungan infrensi dan membership output yang didefinisikan terdapat beberapa metode defuzzifikasi diantaranya Maximum of Mean (MOM), Center of Area (COA) atau center of gravity (COG). Metode COG diskrit sering digunakan untuk defuzzifikasi karena mudah dalam mengimplementasikan dalam bahas a pemrograman. Persamaan defuzzifikasi COG dinyatakan dalam persamaan

|  |  |
| --- | --- |
|  | (22) |

## Fuzzy Logic Control

Pada sub bab ini akan menjelaskan bagaimana implementasi fuzzy untuk permasalaahan control. Sebagai contoh fuzzy digunakan untuk mengontrol kecepatan putar motor DC dimana model motor dc dijabarkan dalam persamaan berikut

|  |  |
| --- | --- |
|  | (23) |
|  | (24) |

Hasil transformasi laplace dari persamaan (23) dan (24)

|  |  |
| --- | --- |
|  | (25) |
|  | (26) |

Untuk menghilangkan i pada model substitusi i pada persamaan (25) ke persamaan (26)

|  |  |
| --- | --- |
|  | (27) |

 Kumpulkan variable sehingga menjadi

|  |  |
| --- | --- |
|  | (28) |

Dari persamaan (28) didapatkan transfer function:

|  |  |
| --- | --- |
|  | (29) |

Dimana:

|  |  |  |
| --- | --- | --- |
| Symbol | keterangan | Nilai parameter |
| (J) | moment of inertia of the rotor  | 0.0167 kg.m^2 |
| (b) | motor viscous friction constant  | 0.0167 N.m.s |
| (Ke) | electromotive force constant  | 0.2 V/rad/sec |
| (Kt) | motor torque constant  | 0.2 N.m/Amp |
| (R) | electric resistance  | 0.6 Ohm |
| (L) | electric inductance  | 0.012 H |
|  | Motor angular velocity | rad/s |

Sehingga didapatkan transfer function pada persamaan (30) berikut:

|  |  |
| --- | --- |
|  | (30) |

Pada model motor tersebut masukan berupa tegangan dan keluaran berupa kecepatan angular motor. Motor tersebut memiliki tegangan kerja maksismum 24 volt dan kecepatan maksimum atau 916 rpm. Pada contoh ini menggunakan struktur fuzzy dengan pendekatan control PD yaitu error dan delta error sebagai masukan system fuzzy yang digambarkan pada blok diagram control fuzzy berikut:



Gambar 3.4 block diagram Fuzzy logic control

Sehingga membership fuzzy input fuzzy terdiri dari dua membership yaitu error dan delta error:



Gambar 3.5 membership function error



Gambar 3.6 membership function delta error

Dengan satu variable membership output yaitu sinyal control u.



Gambar 3.7 membership function sinyal control

Pada input maupun output jumlah membership yaitu 5 buah, sehingga basis aturan dari system fuzzy tersebut memiliki 25 kemungkinan seperti pada table berikut:

Tabel 3.3 Tabel basis aturan fuzzy logic control

|  |  |
| --- | --- |
|  | Error |
| Delta error |  | nb | n | Z | p | pb |
| nb | NB | NB | NB | N | Z |
| n | NB | NB | N | Z | P |
| z | NB | N | Z | P | PB |
| p | N | Z | P | PB | PB |
| pb | Z | P | PB | PB | PB |

Dengan metode infrensi yaitu menggunakan metode Mamdani karena memiliki perihitungan yang lebih mudah serta defuzzifikasi menggunakan metode COG. Sebagai contoh untuk mensimulasikan system fuzzy perhatikan contoh berikut:

Saat waktu t nilai error = 0.06 dan nilai delta error adalah -0.2 maka kita hitung dulu nilai keanggotaan pada masing masing membership function:

|  |  |
| --- | --- |
|   |   |
|   |   |
|  |
|   |   |
|   |   |
|  |

Sehingga & ;

Setelah itu kita hitung nilai keanggotaan berdasarkan rule base menggunakan inferensi Mamdani, diawali dengan menghitung operasi minimum pada nilai keanggotaan input

|  |  |
| --- | --- |
|  | Error |
| Delta error |  | nb(0) | n(0) | z(0.4) | p(0.6) | pb(0) |
| nb (1) | NBmin(0,1)=0 | NBmin(0,1)=0 | NBmin(0.4,1)=0.4 | Nmin(0.6,1)=0.6 | Zmin(0,1)=0 |
| n(0) | NBmin(0,0)=0 | NBmin(0,0)=0 | Nmin(0.4,0)=0 | Zmin(0.6,0)=0 | Pmin(0,0)=0 |
| z(0) | NBmin(0,0)=0 | Nmin(0,0)=0 | Zmin(0.4,0)=0 | Pmin(0.6,0)=0 | PBmin(0,0)=0 |
| p(0) | Nmin(0,0)=0 | Zmin(0,0)=0 | Pmin(0.4,0)=0 | PBmin(0.6,0)=0 | PBmin(0,0)=0 |
| pb(0) | Zmin(0,0)=0 | Pmin(0,0)=0 | PBmin(0.4,0)=0 | PBmin(0.6,0)=0 | PBmin(0,0)=0 |

Menghhitung nilai keanggotaan output dengan operasi max

Sehingga

Selanjutnya proses terakhir pada system fuzzy adalah deffuzifikasi untuk menghitung nilai output nyata:

## Pemrograman Fuzzy pada contoh 3.3

Langkah percobaan:

1. Membuat listing Fuzzifikasi:

|  |
| --- |
| function xf=fuzzifikasi(x,b\_mf)xf=[0,0,0,0,0];mf=[b\_mf(1),b\_mf(2),b\_mf(3),b\_mf(4),b\_mf(5)]; if (x <=mf(1)) xf(1)=1; elseif (mf(1)< x && x<=mf(2)) xf(1)=(mf(2)-x)/(mf(2)-mf(1)); xf(2)=(x-mf(1))/(mf(2)-mf(1)); elseif (mf(2)< x && x<=mf(3)) xf(2)=(mf(3)-x)/(mf(3)-mf(2)); xf(3)=(x-mf(2))/(mf(3)-mf(2)); elseif (mf(3)< x && x<=mf(4)) xf(3)=(mf(4)-x)/(mf(4)-mf(3)); xf(4)=(x-mf(3))/(mf(4)-mf(3)); elseif (mf(4)< x && x<=mf(5)) xf(4)=(mf(5)-x)/(mf(5)-mf(4)); xf(5)=(x-mf(4))/(mf(5)-mf(4)); elseif x >= mf(5) xf(5)=1; endend |

1. Simpan dengan nama fuzzifikasi.m
2. Membuat listing inferensi fuzzy:

|  |
| --- |
| function uU=infuzz(in1,in2)uE=[in1(1),in1(2),in1(3),in1(4),in1(5)];uDE=[in2(1),in2(2),in2(3),in2(4),in2(5)];uU=[0,0,0,0,0];rbf=[1 1 1 2 3; 1 1 2 3 4; 1 2 3 4 5; 2 3 4 5 5; 3 4 5 5 5;]; for i=1:5 for j=1:5 k=rbf(i,j); uU(k)=max(uU(k),min(uE(i),uDE(j))); end endend |

1. Simpan dengan nama infuzz.m
2. Membuat listing defuzzifikasi:

|  |
| --- |
| function u=defuzzifikasi(in)uU=[in(1),in(2),in(3),in(4),in(5)];B=[-24,-12,0,12,24];num=B(1)\*uU(1)+B(2)\*uU(2)+B(3)\*uU(3)+B(4)\*uU(4)+B(5)\*uU(5);denum=uU(1)+uU(2)+uU(3)+uU(4)+uU(5);u=num/denum;end |

1. Simpan dengan nama defuzzifikasi.m
2. Membuat Program utama fuzzy

|  |
| --- |
| function u=fuzzy(in)e=in(1);de=in(2);mE=[-1,-0.1,0,0.1,1];%membership Error mDE=[-1,-0.1,0,0.1,1];%membershib delta Error uE=fuzzifikasi5(e,mE);uDE=fuzzifikasi5(de,mDE);uU=infuz(uE,uDE);u=defuzzifikasi(uU);end |

1. Simpan dengan nama fuzzy.m
2. Run program utama fuzzy.m
3. Abaikan jika muncul hasil running sebagai berikut:



1. Buat Simulink sebagai berikut:



Gambar 3.8 blok Simulink fuzzy logic control pada kecepatan motor

1. Pada blok treansfer fcn double click dan isikan dengan parameter berikut:



1. Click Ok
2. Pada Block *Interpreted MATLAB Function* double click dan isikan dengan parameter berikut:



1. Click Ok
2. Pada block *zero order hold*  double click dan isi time sampling seperti gambar berikut:



1. Click Ok
2. Pada block *unit delay*  double click dan isi time sampling seperti gambar berikut:



1. Click Ok
2. Pada block constant isi dengan nilai 50 seperti gambar berikut:



1. Click ok
2. Run program dan simpan hasil grafik pada kecepatan, sinyal control, error, delta error
3. Pada listing program utama fuzzy rubah batas nilai keanggotaan dengan nilai berikut:



1. Run program kembali dan simpan hasil grafik pada kecepatan, sinyal control, error, delta error
2. Analisa dari perubahan tersebut

## Data Hasil Percobaan

Grafik 3.1 Output Scope Error 0.1

Grafik 3.2 Output Scope Delta Error 0.1

Grafik 3.3 Output Scope Sinyal Control 0.1

Grafik 3.4 Output Scope Kecepatan Angular 0.1

Grafik 3.5 Output Scope Error 0.5

Grafik 3.6 Output Scope Delta Error 0.5

Grafik 3.7 Output Scope Sinyal Control 0.5

Grafik 3.8 Output Scope Kecepatan Angular 0.5

## Analisa Data

## Kesimpulan