

<<COVER JUDUL>>

UU 28 tahun 2014 tentang Hak Cipta

Fungsi dan sifat hak cipta Pasal 4

Hak Cipta sebagaimana dimaksud dalam Pasal 3 huruf a merupakan hak eksklusif yang terdiri atas hak moral dan hak ekonomi.

Pembatasan Perlindungan Pasal 26

Ketentuan sebagaimana dimaksud dalam Pasal 23, Pasal 24, dan Pasal 25 tidak berlaku terhadap:

- a. penggunaan kutipan singkat Ciptaan dan/atau produk Hak Terkait untuk pelaporan peristiwa aktual yang ditujukan hanya untuk keperluan penyediaan informasi aktual;
- b. Penggandaan Ciptaan dan/atau produk Hak Terkait hanya untuk kepentingan penelitian ilmu pengetahuan;
- c. Penggandaan Ciptaan dan/atau produk Hak Terkait hanya untuk keperluan pengajaran, kecuali pertunjukan dan Fonogram yang telah dilakukan Pengumuman sebagai bahan ajar; dan
- d. penggunaan untuk kepentingan pendidikan dan pengembangan ilmu pengetahuan yang memungkinkan suatu Ciptaan dan/atau produk Hak Terkait dapat digunakan tanpa izin Pelaku Pertunjukan, Produser Fonogram, atau Lembaga Penyiaran.

Sanksi Pelanggaran Pasal 113

1. Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf c, huruf d, huruf f, dan/atau huruf h untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 3 (tiga) tahun dan/atau pidana denda paling banyak Rp500.000.000,00 (lima ratus juta rupiah).
2. Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf a, huruf b, huruf e, dan/atau huruf g untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 4 (empat) tahun dan/atau pidana denda paling banyak Rp1.000.000.000,00 (satu miliar rupiah).

PRAKTIK SISTEM PAKAR

Adi Widiatmoko W., S.Si, M.T



STMKG PRESS

SEKOLAH TINGGI METEOROLOGI KLIMATOLOGI DAN GEOFISIKA

PRAKTIK SISTEM PAKAR

Copyright © STMKG Press, 2022

Penulis:

Adi Widiatmoko W.

Editor: ...

Desain Sampul: Tim Kreatif STMKG Press

Sampul: ...

Penerbit

STMKG Press

Web: www.stmkg.ac.id

e-mail: stmkg.press@gmail.com

WA: [08000000111](tel:08000000111)

Adi Widiatmoko W.

Praktik Sistem Pakar

STMKG Press, 2022

xiv; ... hlm; 16 x 23 cm

ISBN: [978-623-6512-68-5](#) (print)

E-ISBN: [978-623-6512-69-2](#) (online)

Cetakan 1, [September](#) 2022

I. Praktik Sistem Pakar

II. STMKG Press

Katalog Dalam Terbitan

Hak cipta dilindungi undang-undang

Dilarang memperbanyak maupun mengedarkan buku tanpa

Ijin tertulis dari penerbit maupun penulis

Cetakan I, [September 2022](#)

KATA PENGANTAR

Berisi tentang pendapat dari orang yang dianggap berkompeten oleh penulis tentang isi dari buku yang ditulis.

PRAKATA

Puji syukur ke hadirat Allah SWT atas rahmat dan karunia-Nya penulis dapat menyelesaikan buku ajar ini yang berjudul Praktik Sistem Pakar. Sholawat dan salam tidak lupa kami ucapkan kepada junjungan kami Rasulullah Muhammad SAW yang senantiasa menginginkan kebaikan bagi seluruh umatnya.

Modul Praktik Sistem Pakar ini disusun sebagai panduan bagi taruna/i dalam mempelajari dan mengembangkan sistem pakar. Sistem pakar merupakan salah satu cabang keilmuan kecerdasan buatan (*artificial intelligence*) yang memungkinkan sistem untuk memecahkan masalah yang kompleks dengan menggunakan pengetahuan dari seorang ahli/pakar di suatu bidang.

Dalam modul praktikum ini, taruna/i akan diberikan kesempatan untuk memahami konsep dasar sistem pakar, termasuk teknik-teknik yang digunakan untuk membangun sistem pakar. Selama praktikum, taruna/i akan membangun sistem pakar sederhana yang dapat membantu dalam memecahkan masalah yang kompleks di bidang yang dipilih.

Kami berharap modul praktikum ini dapat membantu taruna/i untuk memahami konsep dasar sistem pakar, serta memberikan pengalaman praktis dalam membangun sistem pakar sederhana. Dengan demikian, taruna/i diharapkan dapat mengembangkan keterampilan dan pengetahuan yang diperlukan untuk menghadapi tantangan di dunia kerja di masa depan.

Tangerang Selatan, ... September 2022

Adi Widiatmoko W.

DAFTAR ISI

KATA PENGANTAR	v
PRAKATA	vi
DAFTAR ISI	vii
BAB 1 PENDAHULUAN PROLOG	1
A. Tujuan Praktikum	1
B. Pendahuluan	1
C. Terminal Prolog	2
D. Program Prolog	2
E. Menulis <i>output</i>	3
F. Membaca <i>input</i>	4
G. Status <i>output</i> jika ada <i>database</i>	5
H. Laporan	6
BAB 2 <i>LOOPING</i>	7
A. Tujuan Praktikum	7
B. Pendahuluan	7
C. <i>Looping</i> hingga kondisi tertentu	7
D. <i>Looping</i> untuk membaca input	8
E. <i>Looping</i> untuk tampilan menu	8
F. Pencarian <i>database</i> menggunakan <i>bactracking with failure</i>	9
G. Laporan	10
BAB 3 <i>DATABASE</i>	11
A. Tujuan Praktikum	11
B. Pendahuluan	11
C. Menambah klausa pada <i>database</i>	11
D. Menghapus klausa pada <i>database</i>	12
E. Mesin inferensi	13
F. Laporan	14
BAB 4 FUZZY LOGIC	15
A. Tujuan Pembelajaran	15

B.	Pendahuluan	15
C.	Contoh Kasus Pemberian Bonus.....	15
D.	FIS Editor.....	16
E.	Rule Editor.....	20
F.	Rule Viewer dan Surface Viewer	21
G.	Cara Menyimpan File Desain Fuzzy Logic	22
H.	Evaluasi FIS	23
I.	Laporan	23
BAB 5 STUDI KASUS FUZZY LOGIC.....		25
A.	Tujuan Pembelajaran	25
B.	Pendahuluan	25
C.	Studi Kasus Sistem Pengatur Kecepatan Mesin.....	25
D.	FIS Editor.....	27
E.	Rule Editor.....	31
F.	Rule Viewer	32
G.	Evaluasi FIS	34
H.	Laporan	35
DAFTAR PUSTAKA.....		36
HASIL SCANNING SIMILARITY		38
KOMENTAR REVIEWER		40
BIOGRAFI PENULIS		42

BAB 1

PENDAHULUAN PROLOG

A. Tujuan Praktikum

1. Mengoperasikan program Prolog.
2. Menerapkan dasar pemrograman Prolog.

B. Pendahuluan

Prolog adalah bahasa pemrograman berbasis logika atau sering dikenal sebagai *logic programming*. Penyusunan logika dalam pemrogramannya menggunakan metode penghubungan antara fakta dan aturan yang ada dalam database. Bahasa pemrograman Prolog dikembangkan dan diimplementasikan di Marseille, Perancis, pada tahun 1972 oleh Alain Colmerauer dan Philippe Roussel, berdasarkan interpretasi prosedural dari *Horn clauses* milik Robert Kowalski di Universitas Edinburgh. [1][2][3]

Prolog adalah bahasa pemrograman yang paling populer di dunia pendidikan. Hal ini karena Prolog merupakan bahasa pemrograman simbolik yang algoritma dan penulisan programnya relatif lebih sederhana. Para pengguna Prolog dapat lebih fokus pada struktur basis pengetahuan dibandingkan dengan format penulisannya. Di dunia industri perangkat lunak, bahasa pemrograman Prolog juga telah diimplementasikan pada produk-produk Artificial Intelligence seperti IBM Watson [4] dan TerminusDB [5].

Dalam pelaksanaan praktikum ini, dibutuhkan dua *software*, yaitu *software* bahasa pemrograman Prolog untuk menjalankan algoritma sistem pakar, dan *software* editor teks untuk menulis *coding program*. Ada banyak *software* Prolog dan editor teks yang tersedia di internet. Namun jika anda

belum memiliki pilihan, maka berikut adalah dua *software* yang direkomendasikan, yaitu :

1. GNU Prolog (link : <http://www.gprolog.org/>)
Masuk ke situs di atas, lalu *download installer* yang sesuai dengan *bit* sistem operasi PC anda, kemudian lakukan instalasi.
2. Notepad++, (link : <https://notepad-plus-plus.org/>)
Masuk ke situs di atas, lalu *download installer* yang sesuai dengan *bit* sistem operasi PC anda dan lakukan instalasi. Setelah itu, pada Notepad++, lakukan pengaturan bahasa dengan cara mengimpor profil bahasa GNU Prolog yang telah disediakan di situs [gprolog.org](http://www.gprolog.org) atau buka link http://www.gprolog.org/userDefineLang-GNU_Prolog.xml.

C. Terminal Prolog

Praktik

Bukalah GNU Prolog, lalu pada terminal, ketiklah perintah berikut :

```
| ?- write('HelloWorld'),nl,write('Welcome to Prolog'),nl.
```

Tekan Enter dan catatlah respon yang diberikan oleh Prolog. Setiap perintah Prolog wajib diakhiri oleh tanda titik. **nl** adalah singkatan dari *newline*, ia merupakan perintah untuk memulai baris baru.

D. Program Prolog

Praktik

Bukalah Notepad++, lalu tuliskan program berikut :

```
gajah(fido).
```

```
kucing(felix).
```

```
hewan(X):-gajah(X).
```

Simpan teks ini dengan nama *prog1.pl*. Pada terminal Prolog, panggil program tersebut dengan cara klik menu *File > Consult...*, sehingga muncul

notif yang diakhiri dengan kata *yes*. Notifikasi ini berarti bahwa program telah berhasil dipanggil dan siap dijalankan pada terminal.

Ulasan

Berikut adalah penjelasan mengenai makna program di atas.

gajah(fido). adalah sebuah fakta (*fact*)
kucing(felix). adalah sebuah fakta (*fact*)
hewan(X):-gajah(X). adalah sebuah aturan (*rule*)

Setiap fakta adalah klausa (*clause*) yang tersusun dari *predicate(atom)*. Dalam struktur penyusunan logika Prolog, predikat adalah ibarat sebuah wadah dari suatu variabel, nilai, atau konstanta, sementara atom adalah variabel, nilai, atau konstantanya. Dua fakta di atas dapat diterjemahkan sebagai 'Fido adalah gajah' dan 'Felix adalah kucing'.

Aturan (*rule*) adalah klausa yang tersusun dari *head:-body*. Karakter *:-* dapat dibaca sebagai '*if*'. *Head* dan *body* masing-masing adalah fakta yang dirangkai untuk menyusun suatu aturan tertentu. Sebuah atom yang ditulis menggunakan huruf kapital atau teks yang diawali huruf kapital merupakan format penulisan sebuah variabel, misalnya pada contoh di atas, **X** adalah variabel. Dengan kata lain, rules **hewan(X):-gajah(X)** dapat diterjemahkan sebagai X adalah hewan jika X adalah gajah.

Praktik

Selanjutnya, pada terminal, ketiklah perintah berikut :

| ?- **hewan(fido).**

Tekan enter dan catatlah responnya.

| ?- **hewan(felix).**

Tekan enter dan catatlah responnya.

E. Menulis *output*

Praktik

Pada terminal, ketik dan jalankan beberapa perintah berikut :

| ?- write(26).

| ?- write('a string of characters'),nl.

| ?- write([a,b,c,d,[x,y,z]]),nl.

| ?- write(myPred(a,b,c)),nl.

| ?- write('Example of use of nl'),nl,nl,write('end of example'),nl.

Catatlah setiap respon dari Prolog.

Selanjutnya, pada terminal, ketik dan jalankan beberapa perintah berikut :

| ?- writeq('a string of characters'),nl.

| ?- writeq(cat),nl.

| ?- writeq('cat'),nl.

Catatlah setiap respon dari Prolog.

F. Membaca *input*

Praktik

Pada terminal, ketik dan jalankan perintah berikut :

| ?- read(X).

Setelah menekan enter, terminal akan berada dalam status siap menerima *input*, maka berikan *input* :

jim.

Catatlah respon dari Prolog.

Kemudian, pada terminal, lanjutkan dengan beberapa perintah berikut :

```
| ?- read(X).
```

```
26.
```

```
| ?- read(X).
```

```
mypred(a,b,c).
```

```
| ?- read(Z).
```

```
[a,b,mypred(p,q,r),[z,y,x]].
```

```
| ?- read(Y).
```

```
'a string of characters'.
```

Catatlah setiap respon dari Prolog dan berikan analisa mengenai seluruh perintah di atas.

G. Status output jika ada database

Praktik

Pada terminal, ketik dan jalankan dua perintah berikut :

```
| ?- X=fred,read(X).
```

```
jim.
```

```
| ?- X=fred,read(X).
```

```
fred.
```

Catatlah setiap respon dari Prolog dan berikan analisa mengenai dua perintah di atas.

H. Laporan

Buatlah laporan dari kegiatan praktikum di atas dengan format :

1. Pendahuluan (deskripsi kegiatan praktikum yang dilakukan)
2. Hasil dan Pembahasan
3. Kesimpulan

BAB 2

LOOPING

A. Tujuan Praktikum

1. Mengoperasikan cara membuat program *looping* pada Prolog.
2. Menerapkan *looping* untuk berbagai kasus.

B. Pendahuluan

Hampir seluruh bahasa pemrograman memiliki kemampuan untuk melakukan *looping* (pengulangan), begitu juga dengan bahasa Prolog. *Looping* adalah suatu aktivitas program atau perintah yang dieksekusi secara berulang hingga diperoleh kondisi yang diinginkan. Pada bab ini akan dibahas implementasi *looping* pada Prolog yang memiliki metode sedikit berbeda dibanding bahasa pemrograman lain.

C. *Looping* hingga kondisi tertentu

Praktik

Buatlah program berikut :

loop(0).

loop(N):-N>0,write('Nilainya adalah: '),write(N),nl,M is N-1,loop(M).

Simpan program ini dengan nama *loop1.pl*. Pada terminal Prolog, panggil (*consult*) program tersebut dan jalankan dengan mengetik perintah **loop(6)**. Catatlah respon dari Prolog dan berikan analisa cara kerja program tersebut.

Selanjutnya, buatlah program berikut :

```
output_values(Last,Last):-write(Last),nl,write('akhir dari contoh'),nl.
```

```
output_values(First,Last):-First=\=Last,write(First),nl,  
N is First+1,output_values(N,Last).
```

Simpan program ini dengan nama *loop2.pl*. Pada terminal Prolog, panggil program tersebut dan jalankan dengan mengetik perintah **output_values(5,12)**. Catatlah respon dari Prolog dan berikan analisa cara kerja program tersebut.

D. Looping untuk membaca input

Praktik

Buatlah program berikut :

```
go:-loop(start).
```

```
/* start adalah nilai dummy yang digunakan untuk memulai proses  
looping.*/
```

```
loop(end).
```

```
loop(X):-X\=end,write('Ketik sesuatu (atau end untuk mengakhiri): '),  
read(Word),write('Inputnya adalah '),write(Word),nl,loop(Word).
```

Simpan program ini dengan nama *recur1.pl*. Pada terminal Prolog, panggil program tersebut dan jalankan dengan mengetik perintah **go**. Kemudian, lakukan uji coba dengan menginputkan apa saja dan catatlah respon dari Prolog. Berikan analisa cara kerja program tersebut.

E. Looping untuk tampilan menu

Praktik

Buatlah program berikut :

```
go:- nl,write('Berikut adalah cara kerja menu Prolog'),nl,  
menu.
```

```
menu:-nl,write('MENU (pilih salah satu)'),nl,
```

```

write('a. Opsi A'),nl,write('b. Opsi B'),nl,
write('c. Opsi C'),nl,write('d. Akhiri'),nl,
write('Input pilihan: '),read(Input),nl,choice(Input).
choice(a):-write('Opsi A dipilih'),nl,menu.
choice(b):-write('Opsi B dipilih'),nl,menu.
choice(c):-write('Opsi C dipilih'),nl,menu.
choice(d):-write('Sampai jumpa!'),nl.
choice(_):-write('Pilihan tidak tersedia, silakan coba lagi'),nl,menu.

```

Simpan program ini dengan nama *menu1.pl*. Pada terminal Prolog, panggil program tersebut dan jalankan dengan mengetik perintah **go**. Kemudian, lakukan uji coba dengan menginputkan apa saja dan catatlah respon dari Prolog. Berikan analisa cara kerja program tersebut.

F. Pencarian *database* menggunakan *backtracking with failure*

Praktik

Buatlah suatu program dengan *database* seperti berikut :

```

cat(tom).
cat(felix).
cat(garfield).

```

```

allcats:-cat(X),write(X),write(' adalah kucing'),nl,fail.
allcats.

```

Simpan program dengan nama *search1.pl*. Pada terminal Prolog, panggil program tersebut dan jalankan dengan mengetik perintah **allcats**. Catatlah respon dari Prolog.

Kemudian edit program *search1.pl* dengan menghapus perintah **allcats** di baris paling akhir, sehingga programnya menjadi :

```

cat(tom).

```

cat(felix).

cat(garfield).

allcats:-cat(X),write(X),write(' adalah kucing'),nl,fail.

Simpan program lalu pada terminal Prolog, panggil ulang program tersebut dan ketik perintah **allcats**. Catatlah perbedaan yang terjadi pada respon Prolog. Berikan analisa mengenai kedua format penulisan program di atas.

Selanjutnya, buatlah suatu program dengan *database* seperti berikut :

person(john,smith,45,london,dokter).

person(martin,williams,33,birmingham,guru).

person(henry,smith,26,manchester,teknisi).

person(jane,wilson,62,london,guru).

person(mary,smith,29,glasgow,manajer).

**allteachers:-person(Forename,Surname,_,_guru),write(Forename),
write(' '),write(Surname),nl,fail.**

allteachers.

Simpan program dengan nama *search2.pl*. Pada terminal Prolog, panggil program tersebut dan jalankan dengan mengetik perintah **allteachers**. Catatlah respon dari Prolog dan berikan analisa cara kerja program tersebut.

G. Laporan

Buatlah laporan dari kegiatan praktikum di atas dengan format :

1. Pendahuluan (deskripsi kegiatan praktikum yang dilakukan)
2. Hasil dan Pembahasan
3. Kesimpulan

BAB 3

DATABASE

A. Tujuan Praktikum

1. Mengoperasikan cara mengubah *database* pada Prolog.
2. Menerapkan perubahan *database* untuk berbagai kasus.

B. Pendahuluan

Pada Prolog, setelah melakukan pemanggilan (*consult*) program, maka *database* yang ada dalam program tersebut akan tetap aktif pada terminal hingga dilakukan pemanggilan ulang atau *me-restart* aplikasi Prolog. Cara ini merupakan cara paling dasar untuk melakukan perubahan *database*. Namun, jika kita hendak membuat suatu program yang bisa aktif menambah dan menghapus *database* meskipun dalam kondisi sedang aktif di terminal, Prolog telah menyediakan *built-in predicate* untuk melakukan hal tersebut.

C. Menambah klausa pada *database*

Praktik

Pada terminal, ketiklah perintah berikut :

```
| ?- listing.
```

Catatlah respon dari Prolog. Lalu, ketiklah perintah berikut :

```
| ?- assertz(gajah(fido)).
```

```
| ?- assertz((go:-write('hello world'),nl)).
```

```
| ?- listing.
```

Catatlah respon dari Prolog.

Selanjutnya, pada terminal, ketiklah perintah berikut :

```
| ?- asserta(gajah(jumbo)).
```

```
| ?- asserta((go:-write('thank you'),nl)).
```

```
| ?- listing.
```

Catatlah respon dari Prolog dan berikan analisa cara kerja *built-in predicate* `assertz` dan `asserta`.

D. Menghapus klausa pada *database*

Praktik

Buatlah suatu program dengan *database* seperti berikut :

```
:-dynamic(gajah/1).
```

```
gajah(jim).
```

```
gajah(fido).
```

```
gajah(henry).
```

```
gajah(X).
```

Simpan program dengan nama *database1.pl*. Pada terminal Prolog, panggil program tersebut, lalu ketiklah perintah **listing** dan catat respon dari Prolog. Setelah itu, pada terminal, ketik perintah berikut :

```
| ?- retract(gajah(fido)).
```

Catat respon Prolog, lalu ketik lagi perintah **listing**. Catatlah perubahan yang terjadi pada respon Prolog dan berikan analisa cara kerja *built-in predicate* `retract`.

Selanjutnya, pada terminal, ketik perintah berikut :

```
| ?- retract(gajah(X)).
```

Catat respon Prolog, lalu ketik lagi perintah **listing**. Catatlah respon Prolog dan berikan analisa cara kerja perintah **retract** variabel di atas.

Masih menggunakan program *database1.pl*, panggil program tersebut pada terminal, lalu ketiklah perintah **listing** dan catat respon dari Prolog. Setelah itu, pada terminal, ketik perintah berikut :

```
| ?- retractall(gajah(_)).
```

Catat respon Prolog, lalu ketik lagi perintah **listing**. Catatlah perubahan yang terjadi pada respon Prolog dan berikan analisa cara kerja *built-in predicate* **retractall**.

E. Mesin inferensi

Praktik

Buatlah suatu program dengan *database* seperti berikut :

```
bird(laysan_albatross):-family(albatross),color(white).
```

```
bird(black_footed_albatross):-family(albatross),color(dark).
```

```
bird(whistling_swan):-family(swan),voice(muffled_musical_whistle).
```

```
bird(trumpeter_swan):-family(swan),voice(loud_trumpeting).
```

```
mulai:-retractall(family(_)),retractall(color(_)),retractall(voice(_)),nl,  
write('===SISTEM PAKAR IDENTIFIKASI HEWAN==='),nl,  
menu1.
```

```
menu1:-nl,write('Pilih ciri hewan'),nl,  
write('Family:'),nl,  
write('a. albatross'),nl,  
write('b. swan'),nl,  
write('Input pilihan: '),read(Family),opsi1(Family).
```

```
opsi1(a):-assertz(family(albatross)),menu2.
```

```
opsi1(b):-assertz(family(swan)),menu3.
```

```
menu2:-nl,write('Color:'),nl,  
write('a. white'),nl,  
write('b. dark'),nl,  
write('Input pilihan: '),read(Color),opsi2(Color).
```

```
opsi2(a):-assertz(color(white)),solusi.  
opsi2(b):-assertz(color(dark)),solusi.
```

```
menu3:-nl,write('Voice:'),nl,  
write('a. muffled musical whistle'),nl,  
write('b. loud trumpeting'),nl,  
write('Input pilihan: '),read(Voice),opsi3(Voice).
```

```
opsi3(a):-assertz(voice(muffled_musical_whistle)),solusi.  
opsi3(b):-assertz(voice(loud_trumpeting)),solusi.
```

```
solusi:-bird(X),nl,write('Jenis hewan: '),write(X),nl,  
nl,write('-----Program Selesai-----').
```

Simpan program ini dengan nama *inference1.pl*. Pada terminal Prolog, panggil program tersebut dan jalankan dengan mengetik perintah **mulai**. Kemudian, lakukan uji coba dengan menginputkan apa saja dan catatlah respon dari Prolog. Berikan analisa cara kerja program tersebut.

F. Laporan

Buatlah laporan dari kegiatan praktikum di atas dengan format :

1. Pendahuluan (deskripsi kegiatan praktikum yang dilakukan)
2. Hasil dan Pembahasan
3. Kesimpulan

BAB 4

FUZZY LOGIC

A. Tujuan Pembelajaran

1. Mengoperasikan FIS Editor Matlab.
2. Menerapkan konsep dasar *fuzzy logic* pada FIS Editor.

B. Pendahuluan

Fuzzy logic telah banyak digunakan pada berbagai aplikasi mulai dari produk konsumen seperti kamera, mesin cuci, dan microwave hingga kontrol proses industri, instrumentasi medis, dan sistem pendukung keputusan. Dalam perkembangannya, penggunaan *fuzzy logic* telah dikombinasikan dengan *neurocomputing* dan *genetic algorithm* yang dikenal dengan istilah *soft computing* [6] [7].

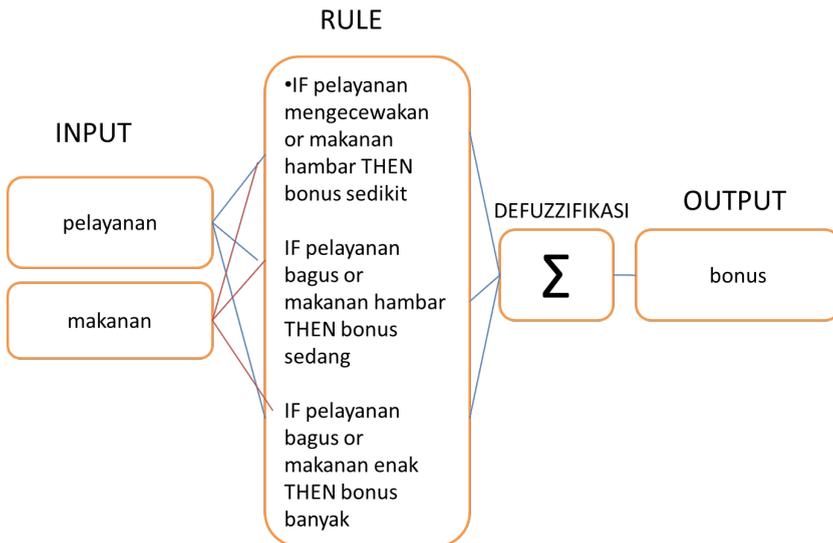
Dalam pelaksanaan praktikum ini, dibutuhkan *software* komputasi *fuzzy logic*. Ada banyak *software fuzzy logic* yang tersedia di internet. Namun jika anda belum memiliki pilihan, maka *software* yang direkomendasikan adalah Fuzzy Logic Toolbox pada Matlab.

C. Contoh Kasus Pemberian Bonus

Diberikan contoh kasus ilustrasi yaitu hubungan pelayanan restoran dengan pemberian bonus kepada pelayan. Misalkan hubungan bonus dengan pelayanan dirumuskan sebagai berikut :

- Pelayanan diberi skor 0 – 10, di mana skor 0 adalah buruk dan skor 10 adalah memuaskan.
- Masakan diberi skor 0 – 10, di mana skor 0 adalah hambar dan skor 10 adalah enak.

Permasalahan selanjutnya adalah cara menentukan besar bonus untuk seorang pelayan restoran berdasarkan inputan pelayanan dan masakan dengan rentang 0 – 10. Dalam diagram sistem *fuzzy logic*, kasus di atas dapat digambarkan sebagai berikut.

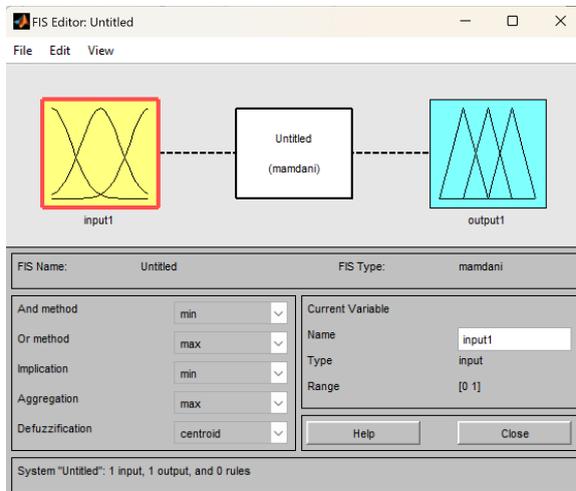


Gambar 4.1 Diagram sistem *fuzzy logic*.

Selanjutnya misalkan suatu fuzzifikasi sebagai berikut. Pelayanan dibagi menjadi tiga fuzzy set, yaitu mengecewakan, bagus, dan memuaskan. Masakan dibagi menjadi dua fuzzy set, yaitu hambar dan enak. Bonus dibagi menjadi tiga fuzzy set yaitu sedikit, sedang, dan banyak.

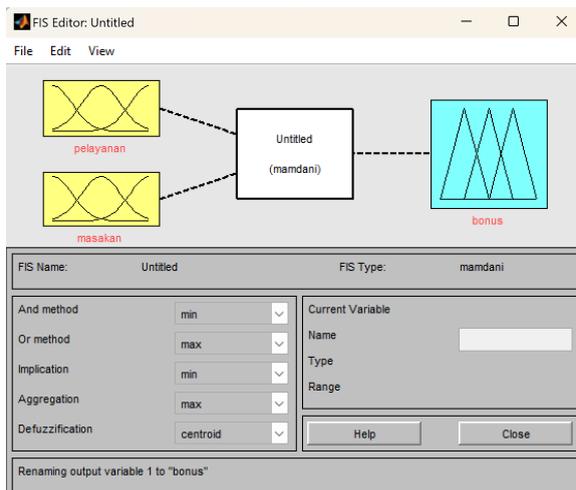
D. FIS Editor

Pada Matlab, bukalah FIS Editor dengan cara mengetik perintah “*fuzzy*” pada Command Window atau melalui menu Apps > Fuzzy Logic Design (Matlab R2013b).



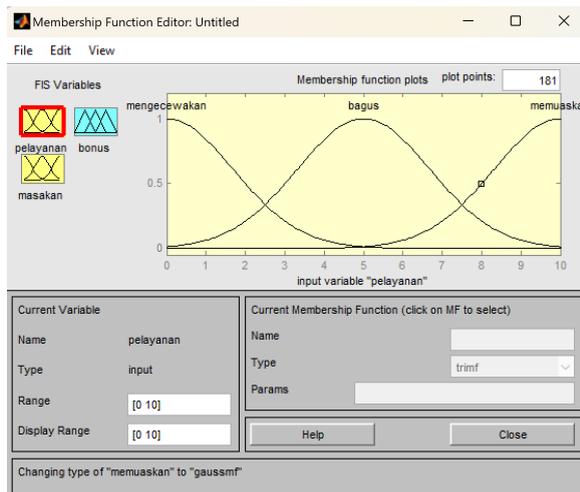
Gambar 4.2 Tampilan FIS Editor.

Secara otomatis (default), Matlab menyediakan satu input, satu output dan satu rule bertipe Mamdani. Namun berdasarkan rumusan sistem *fuzzy logic* di atas, kita akan membuat dua input yaitu pelayanan dan masakan. Caranya adalah dengan klik menu Edit > Add Variable... > Input. Selanjutnya, beri nama “pelayanan” dan “masakan” sebagai variabel input. Lalu, beri nama yang sesuai untuk output, yaitu “bonus”.



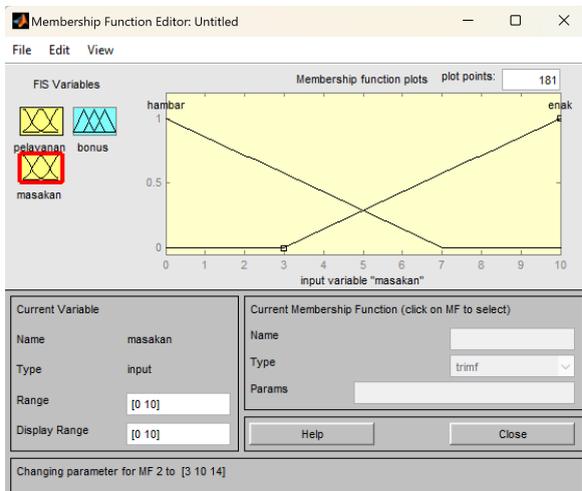
Gambar 4.3 Tampilan FIS Editor setelah nama variabel berhasil diubah.

Untuk pendefinisian variabel pelayanan, klik ikon variabel FIS pelayanan. Pada field Range, ubah menjadi [0 10]. Pada field Display Range, ubah menjadi [0 10]. Lalu, klik kurva berlabel **mf1**. Pada field Name ubah menjadi “mengecewakan”. Pada field Type pilih opsi **gaussmf**. Pada field Params dibiarkan default. Lakukan juga langkah-langkah tersebut pada kurva **mf2**, diganti dengan “bagus”, dan **mf3** diganti dengan “memuaskan”.



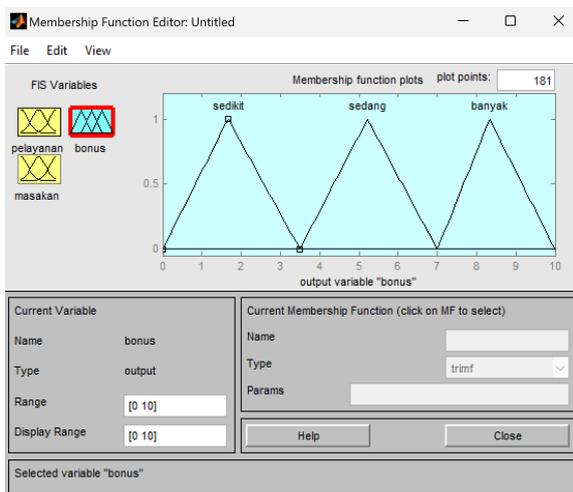
Gambar 4.4 Tampilan Membership Function Editor variabel FIS pelayanan.

Untuk pendefinisian variabel masakan, klik ikon variabel FIS masakan. Lalu klik kurva berlabel **mf2**, pilih menu Edit > Remove Selected MF, sehingga **mf2** terhapus. Pada field Range, ubah menjadi [0 10]. Pada field Display Range ubah menjadi [0 10]. Selanjutnya, klik kurva berlabel **mf1**, pada field Name ubah menjadi “hambar”. Pada field Type pilih opsi **trimf**. Pada field Params diubah menjadi [-4 0 7] di mana nilai pada kolom pertama dapat diabaikan karena di luar rentang harga variabel masakan. Lakukan langkah yang sama pada kurva **mf3**, diganti menjadi “enak” dan pada field Params diubah menjadi [3 10 14] di mana nilai pada kolom ketiga dapat diabaikan karena ia juga di luar rentang harga variabel masakan.



Gambar 4.5 Tampilan Membership Function Editor variabel FIS masakan.

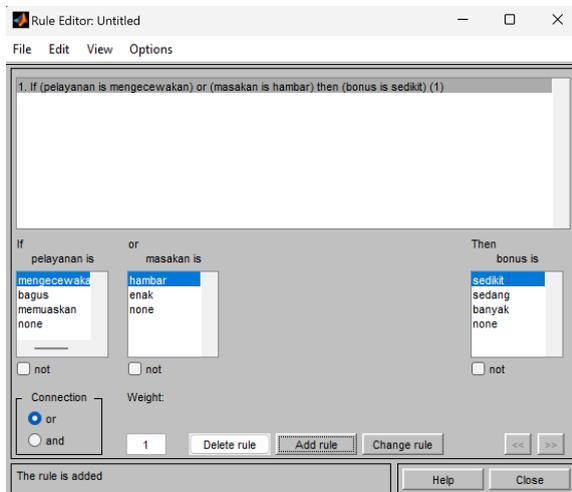
Untuk pendefinisian variabel bonus, klik ikon variabel FIS bonus. Lalu, pada field Range ubah menjadi [0 10] dan pada field Display Range ubah menjadi [0 10]. Selanjutnya, klik kurva berlabel **mf1**. Pada field Name ubah menjadi “sedikit”. Pada field Type pilih opsi **trimf**. Pada field Params dibuat [0 1.667 3.5]. Lakukan langkah yang sama pada kurva **mf2**, diganti dengan “sedang”, pada field Params dibuat [3.5 5.222 7] dan pada kurva **mf3** diganti dengan “banyak”, pada field Params dibuat [7 8.333 10].



Gambar 4.6 Tampilan Membership Function Editor variabel FIS bonus.

E. Rule Editor

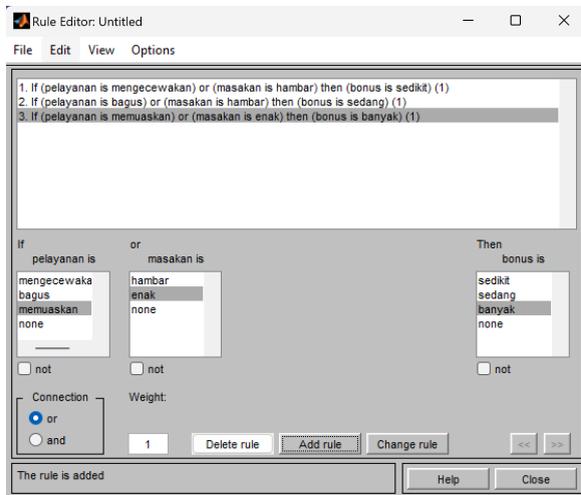
Setelah itu, kita kembali ke jendela utama FIS Editor. Doppel klik pada gambar kotak tengah FIS Editor untuk membuka jendela Rule Editor. Pilih opsi logika yang ingin didefinisikan, misalnya pada variabel pelayanan, pilih “mengecewakan”. Pada variabel masakan, pilih “hambar”. Di bagian blok Connection, pilih **or**. Pada variabel bonus, pilih “sedikit”. Pada field Weight, set bobot rule menjadi 1 (default). Lalu, klik Add rule. Pada tahapan ini, kita telah membuat satu rule, yaitu *if (pelayanan is mengecewakan) or (masakan is hambar) then (bonus is sedikit)*.



Gambar 4.7 Tampilan Rule Editor setelah diberikan satu rule.

Kemudian, dengan langkah yang sama, lanjutkan membuat rule berikut :

- *if (pelayanan is bagus) or (masakan is hambar) then (bonus is sedang)*
- *if (pelayanan is memuaskan) or (masakan is enak) then (bonus is banyak)*

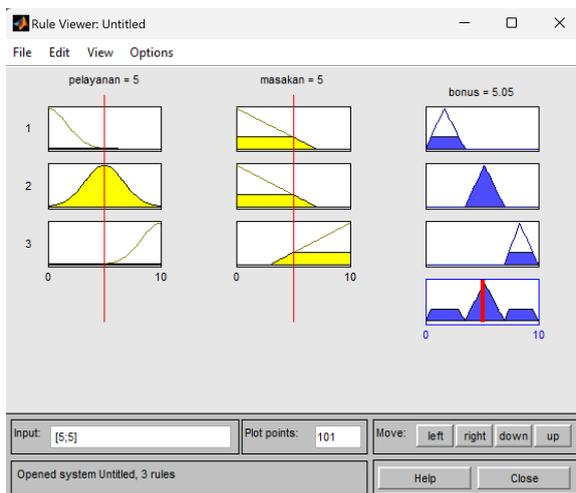


Gambar 4.8 Tampilan Rule Editor setelah diberikan tiga rule.

Jika suatu saat hendak mengedit sebuah rule, maka klik rule yang ingin diedit. Lalu, lakukan perubahan yang diinginkan dan klik Change rule.

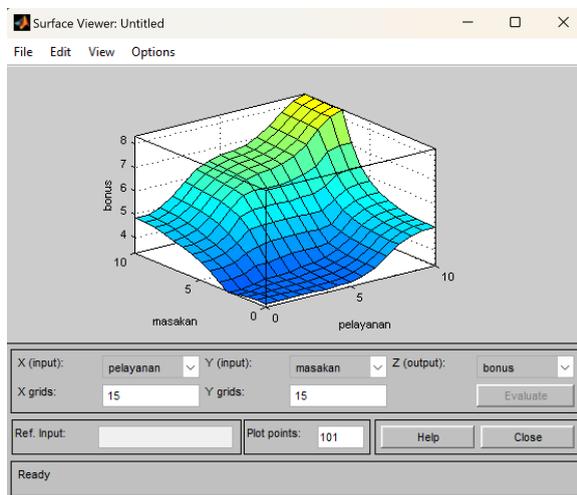
F. Rule Viewer dan Surface Viewer

Untuk membuka jendela Rule Viewer, klik menu View > Rules. Rule viewer berfungsi untuk menampilkan proses perhitungan yang terjadi dalam FIS.



Gambar 4.9 Tampilan Rule Viewer.

Kolom pertama, kedua dan ketiga merepresentasikan variabel pelayanan, masakan, dan bonus. Kita bisa mengubah nilai atau menggeser garis vertikal merah pada masing-masing plot, sehingga nilai variabel yang sesuai akan ditampilkan pada field Input. Jika ingin mengubah nilai inputan secara manual, maka editlah nilai di field Input lalu tekan Enter. Gambar plot di baris keempat adalah plot hasil agregasi, sedangkan hasil defuzzifikasi ditampilkan pada 3 plot bagian atas kolom ketiga. Hasil defuzzifikasi akan berubah sesuai dengan perubahan harga variabel masukan. Untuk melihat hasil FIS dalam plot 3 dimensi, maka klik menu View > Surface.



Gambar 4.10 Tampilan Surface Viewer.

G. Cara Menyimpan File Desain Fuzzy Logic

FIS yang telah dibuat dapat disimpan dalam ekstensi *.fis. Caranya menyimpan file FIS adalah dengan klik menu File > Export > To File..., dan untuk membuka file FIS gunakan menu File > Import > From File.... Pada menu Export dan Import, terdapat pilihan Workspace. Pilihan ini hanya digunakan untuk mengeksport atau mengimpor desain FIS antara Editor dan Workspace Matlab yang sedang aktif saja. Jika Matlab di-shutdown, data akan hilang. Oleh karena itu, lebih baik menyimpannya dalam bentuk file. Misalnya ketika menyimpan desain FIS, setelah klik To file, beri nama file

misalnya bonus. File desain fuzzy logic akan tersimpan dengan nama dan ekstensi “bonus.fis”.

H. Evaluasi FIS

Tahap berikutnya adalah menguji fuzzy logic yang telah dibuat. Caranya adalah dengan kembali ke command window, lalu ketik :

```
>> fis = readfis('bonus')
```

Tekan enter, maka Matlab akan me-load engine FIS yang telah kita desain pada file bonus.fis. Jika gagal, pastikan folder kerja (Current Folder) yang aktif adalah lokasi folder tempat kita menyimpan file bonus.fis sebelumnya. Berikutnya kita uji misalnya pelayanan buruk dengan skor [1] dan masakannya pun tidak enak dengan skor [1]. Instruksi yang digunakan adalah evalfis.

```
>> hasil = evalfis([1 1],fis)
```

Tekan enter, maka akan muncul hasil berikut.

```
hasil =
```

```
3.4823
```

Angka 3,4823 di atas adalah skor bonus yang diperoleh dari hasil perhitungan engine FIS yang kita gunakan. Ingat bahwa skala yang dipakai adalah 0 – 10 sehingga skor 3,4823 menunjukkan bahwa hasil/bonusnya adalah sedikit.

I. Laporan

Buatlah laporan dari kegiatan praktikum di atas dengan format :

1. Pendahuluan (deskripsi kegiatan praktikum yang dilakukan)

2. Hasil dan Pembahasan
3. Kesimpulan

BAB 5

STUDI KASUS FUZZY LOGIC

A. Tujuan Pembelajaran

1. Membuat desain *fuzzy logic* menggunakan FIS Editor Matlab.
2. Menerapkan desain *fuzzy logic* pada studi kasus.

B. Pendahuluan

Pada bab sebelumnya, kita telah belajar cara mengoperasikan FIS Editor Matlab dan mencobanya pada contoh kasus dasar dari desain *fuzzy logic*. Selanjutnya, kita akan mencoba melakukan studi kasus penerapan desain *fuzzy logic* di bidang instrumentasi agar taruna/i semakin fasih dalam membuat mesin inferensi *fuzzy logic* di lapangan.

C. Studi Kasus Sistem Pengatur Kecepatan Mesin

Misalkan suatu sistem pengatur kecepatan mesin menggunakan masukan dari sensor suhu dan sensor cahaya. Untuk sensor suhu (input1) dibagi menjadi 5 kategori, yaitu :

Tabel 5.1 Kategori Suhu

Kategori	Rentang Suhu
Dingin	0 – 15 °C
Sejuk	11 – 25 °C
Normal	21 – 30 °C
Hangat	28 – 40 °C
Panas	36 – 50 °C

Untuk sensor cahaya dibagi menjadi 3 kategori, yaitu :

Tabel 5.2 Kategori Cahaya

Kategori	Rentang Intensitas Cahaya
Gelap	0 – 35 Cd
Normal	31 – 85 Cd
Terang	81 – 100 Cd

Dan kecepatan putaran mesin dibagi menjadi 3 kategori, yaitu :

Tabel 5.3 Kategori Putaran Mesin

Kategori	Rentang Kecepatan Putaran Mesin
Lambat	0 – 15 m/s
Sedang	11 – 21 m/s
Cepat	19 – 45 m/s

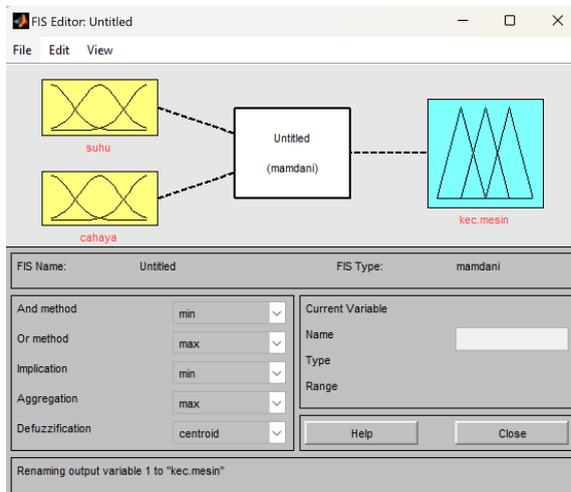
Selanjutnya dibuat rules sistem kontrol sebagai berikut :

Tabel 5.4 Rules Sistem Pengatur Kecepatan Mesin

No.	Input		Output
	Suhu	Cahaya	Kecepatan Mesin
1.	Dingin	Gelap	Lambat
2.	Dingin	Normal	Lambat
3.	Dingin	Terang	Lambat
4.	Sejuk	Gelap	Lambat
5.	Sejuk	Normal	Lambat
6.	Sejuk	Terang	Sedang
7.	Normal	Gelap	Sedang
8.	Normal	Normal	Sedang
9.	Normal	Terang	Sedang
10.	Hangat	Gelap	Sedang
11.	Hangat	Normal	Cepat
12.	Hangat	Terang	Cepat
13.	Panas	Gelap	Cepat
14.	Panas	Normal	Cepat
15.	Panas	Terang	Cepat

D. FIS Editor

Pada Matlab, bukalah FIS Editor dengan cara mengetik perintah “fuzzy” pada Command Window. Berdasarkan rumusan sistem *fuzzy logic* di atas, kita akan membuat dua input yaitu suhu dan cahaya. Caranya adalah dengan klik menu Edit > Add Variable... > Input. Selanjutnya, beri nama “suhu” dan “cahaya” sebagai variabel input. Lalu, beri nama yang sesuai untuk output, yaitu “kec.mesin”.

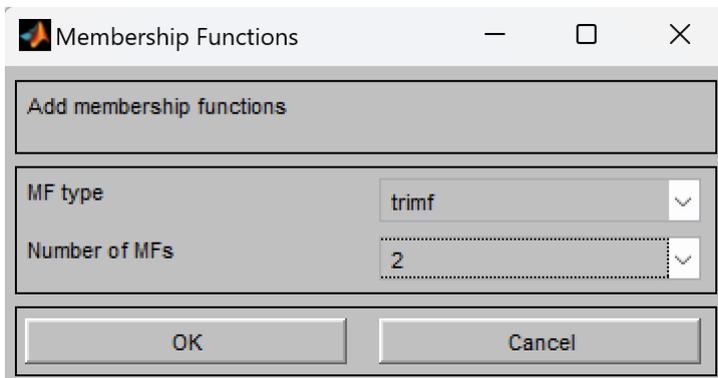


Gambar 5.1 Tampilan FIS Editor setelah nama variabel diubah.

Untuk pendefinisian variabel pelayanan, klik ikon variabel FIS suhu. Pada field Range, ubah menjadi [0 50]. Pada field Display Range, ubah menjadi [0 50]. Lalu, lakukan pengaturan kurva sebagai berikut :

1. Klik kurva berlabel **mf1**. Pada field Name, ubah menjadi “dingin”. Pada field Type pilih opsi **trimf**. Pada field Params, ubah menjadi [0 0 15].

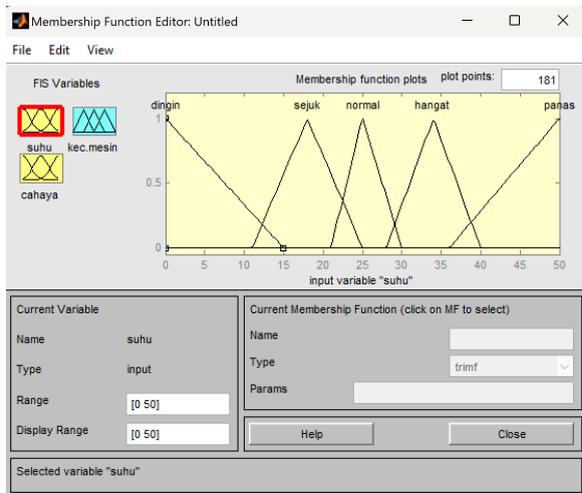
2. Klik kurva berlabel **mf2**. Pada field Name, ubah menjadi “sejuk”. Pada field Type pilih opsi **trimf**. Pada field Params, ubah menjadi [11 18 25].
3. Klik kurva berlabel **mf3**. Pada field Name, ubah menjadi “normal”. Pada field Type pilih opsi **trimf**. Pada field Params, ubah menjadi [21 25 30].
4. Untuk melengkapi lima keadaan variabel input suhu, tambahkan dua kurva membership function dengan cara klik menu Edit > MFs..., dan isikan jumlah kurva yang ingin ditambahkan seperti pada Gambar 5.2.



Gambar 5.2 Menu penambahan kurva membership function.

5. Selanjutnya, klik kurva berlabel **mf4**. Pada field Name, ubah menjadi “hangat”. Pada field Type pilih opsi **trimf**. Pada field Params, ubah menjadi [28 34 40].
6. Klik kurva berlabel **mf5**. Pada field Name, ubah menjadi “panas”. Pada field Type pilih opsi **trimf**. Pada field Params, ubah menjadi [36 50 50].

Jika pengaturan pada semua kurva sesuai petunjuk di atas, tampilannya akan menjadi seperti pada gambar 5.3.

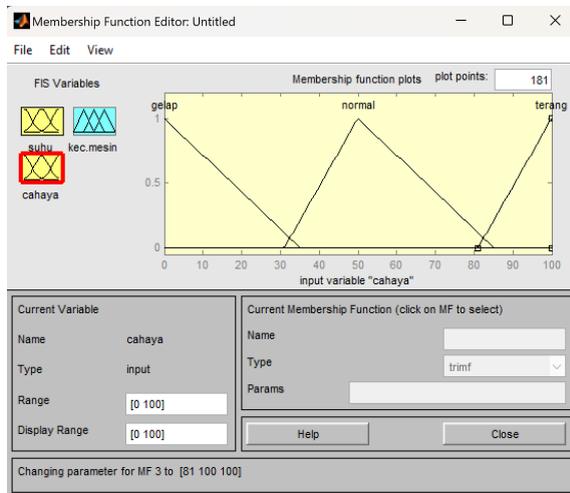


Gambar 5.3 Tampilan Membership Function Editor variabel FIS suhu.

Untuk pendefinisian variabel cahaya, klik ikon variabel FIS cahaya. Pada field Range, ubah menjadi [0 100]. Pada field Display Range ubah menjadi [0 100]. Lalu, lakukan pengaturan kurva sebagai berikut :

1. Klik kurva berlabel **mf1**. Pada field Name, ubah menjadi “gelap”. Pada field Type pilih opsi **trimf**. Pada field Params, ubah menjadi [0 0 35].
2. Klik kurva berlabel **mf2**. Pada field Name, ubah menjadi “normal”. Pada field Type pilih opsi **trimf**. Pada field Params, ubah menjadi [31 50 85].
3. Klik kurva berlabel **mf3**. Pada field Name, ubah menjadi “terang”. Pada field Type pilih opsi **trimf**. Pada field Params, ubah menjadi [81 100 100].

Jika pengaturan pada semua kurva sesuai petunjuk di atas, tampilannya akan menjadi seperti pada gambar 5.4.

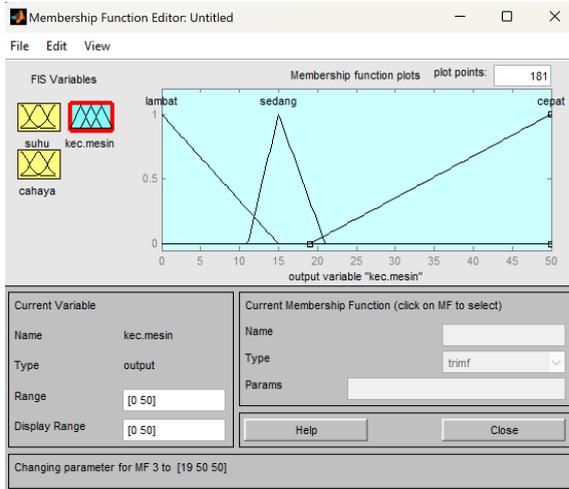


Gambar 5.4 Tampilan Membership Function Editor variabel FIS cahaya.

Untuk pendefinisian variabel kecepatan mesin, klik ikon variabel FIS kec.mesin. Pada field Range, ubah menjadi [0 50]. Pada field Display Range ubah menjadi [0 50]. Lalu, lakukan pengaturan kurva sebagai berikut :

1. Klik kurva berlabel **mf1**. Pada field Name, ubah menjadi “lambat”. Pada field Type pilih opsi **trimf**. Pada field Params, ubah menjadi [0 0 15].
2. Klik kurva berlabel **mf2**. Pada field Name, ubah menjadi “sedang”. Pada field Type pilih opsi **trimf**. Pada field Params, ubah menjadi [11 15 21].
3. Klik kurva berlabel **mf3**. Pada field Name, ubah menjadi “cepat”. Pada field Type pilih opsi **trimf**. Pada field Params, ubah menjadi [19 50 50].

Jika pengaturan pada semua kurva sesuai petunjuk di atas, tampilannya akan menjadi seperti pada gambar 5.5.



Gambar 5.5 Tampilan Membership Function Editor variabel FIS kec.mesin.

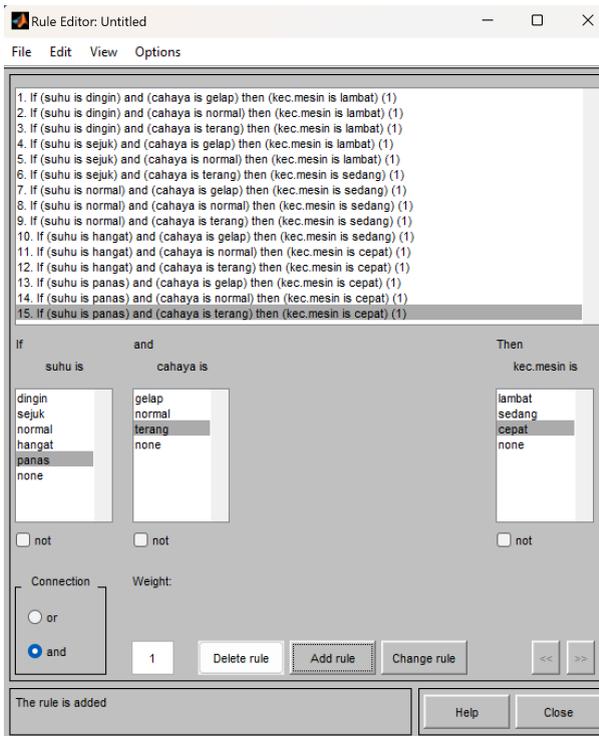
E. Rule Editor

Setelah itu, kita kembali ke jendela utama FIS Editor. Double klik pada gambar kotak tengah FIS Editor untuk membuka jendela Rule Editor. Lakukan pendefinisian logika sistem kontrol sesuai Tabel 5.4 dengan pembobotan dibiarkan default (1). Jika dirincikan, daftar rule-nya adalah sebagai berikut :

1. if (suhu is dingin) and (cahaya is gelap) then (kec.mesin is lambat).
2. if (suhu is dingin) and (cahaya is normal) then (kec.mesin is lambat).
3. if (suhu is dingin) and (cahaya is terang) then (kec.mesin is lambat).
4. if (suhu is sejuk) and (cahaya is gelap) then (kec.mesin is lambat).
5. if (suhu is sejuk) and (cahaya is normal) then (kec.mesin is lambat).
6. if (suhu is sejuk) and (cahaya is terang) then (kec.mesin is sedang).
7. if (suhu is normal) and (cahaya is gelap) then (kec.mesin is sedang).
8. if (suhu is normal) and (cahaya is normal) then (kec.mesin is sedang).
9. if (suhu is normal) and (cahaya is terang) then (kec.mesin is sedang).
10. if (suhu is hangat) and (cahaya is gelap) then (kec.mesin is sedang).

11. if (suhu is hangat) and (cahaya is normal) then (kec.mesin is cepat).
12. if (suhu is hangat) and (cahaya is terang) then (kec.mesin is cepat).
13. if (suhu is panas) and (cahaya is gelap) then (kec.mesin is cepat).
14. if (suhu is panas) and (cahaya is normal) then (kec.mesin is cepat).
15. if (suhu is panas) and (cahaya is terang) then (kec.mesin is cepat).

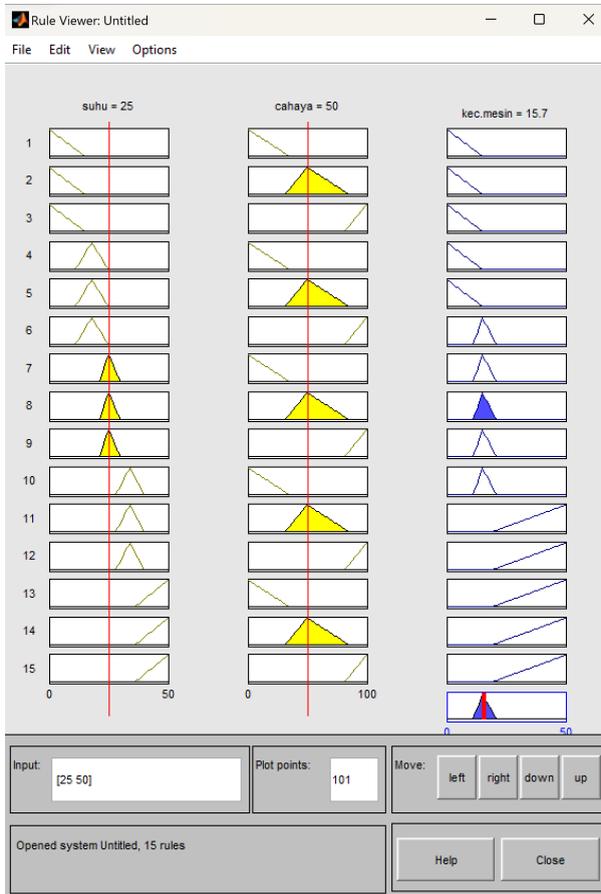
Jika pendefinisian rule sesuai petunjuk di atas, tampilannya akan menjadi seperti pada gambar 5.6.



Gambar 5.6 Tampilan Rule Editor setelah diberikan rule sistem kontrol.

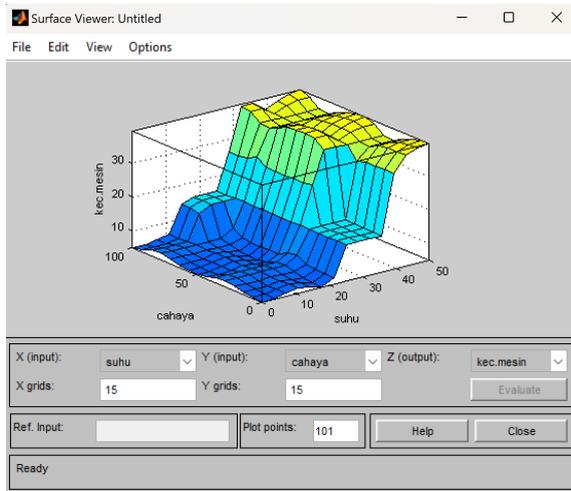
F. Rule Viewer

Untuk mengecek proses perhitungan yang terjadi dalam FIS, dapat dilakukan dengan membuka jendela Rule Viewer, klik menu View > Rules.



Gambar 5.7 Tampilan Rule Viewer sistem kontrol.

Seperti pada Gambar 5.7, kolom pertama, kedua dan ketiga merepresentasikan variabel suhu, cahaya, dan kecepatan mesin. Kita bisa mengecek proses perhitungan FIS dengan mengubah nilai atau menggeser garis vertikal merah pada masing-masing plot, sehingga nilai variabel yang sesuai akan ditampilkan pada field Input. Gambar plot di baris paling bawah (baris ke-16) adalah plot hasil agregasi, sedangkan hasil defuzzifikasi ditampilkan pada plot di atasnya. Untuk melihat hasil FIS dalam plot 3 dimensi, maka klik menu View > Surface.



Gambar 5.8 Tampilan Surface Viewer sistem kontrol.

FIS yang telah dibuat dapat disimpan dengan cara klik menu File > Export > To File..., beri nama file misalnya “kecepatanmesin”. File desain fuzzy logic akan tersimpan dengan nama “kecepatanmesin.fis”.

G. Evaluasi FIS

Tahap berikutnya adalah menguji fuzzy logic yang telah dibuat. Caranya adalah dengan kembali ke command window, lalu ketik :

```
>> fis = readfis('kecepatanmesin')
```

Tekan enter, maka Matlab akan me-load engine FIS yang telah kita desain pada file kecepatanmesin.fis. Berikutnya kita uji misalnya dengan instruksi berikut.

```
>> hasil = evalfis([10 20],fis)
```

Tekan enter, maka akan muncul hasil berikut.

hasil =

6.2059

Angka 6,2059 di atas adalah skor kecepatan mesin yang diperoleh dari hasil perhitungan engine FIS yang kita gunakan. Skala yang kita pakai adalah 0 – 50 sehingga skor 6,2059 menunjukkan bahwa hasil kecepatan mesinnya adalah lambat.

H. Laporan

Buatlah laporan dari kegiatan praktikum di atas dengan format :

1. Pendahuluan (deskripsi kegiatan praktikum yang dilakukan)
2. Hasil dan Pembahasan
3. Kesimpulan

DAFTAR PUSTAKA

- [1] Kowalski, R. A. (1988). "The early years of logic programming" (PDF). *Communications of the ACM*. 31: 38. doi:10.1145/35043.35046. S2CID 12259230.
- [2] Colmerauer, A.; Roussel, P. (1993). "The birth of Prolog" (PDF). *ACM SIGPLAN Notices*. 28 (3): 37. doi:10.1145/155360.155362.
- [3] "PROLOG:a brief history". Retrieved 21 November 2021.
- [4] Adam Lally; Paul Fodor (31 March 2011). "Natural Language Processing With Prolog in the IBM Watson System". Association for Logic Programming. See also Watson (computer).
- [5] <https://github.com/terminusdb/terminusdb>, TerminusDB, 2022-08-24, retrieved 2022-08-24
- [6] Choudhury, Balamati; Jha, Rakesh Mohan, eds. (2016), "Soft Computing Techniques", *Soft Computing in Electromagnetics: Methods and Applications*, Cambridge: Cambridge University Press, pp. 9–44, doi:10.1017/CBO9781316402924.003, ISBN 978-1-107-12248-2, retrieved 2021-02-24
- [7] Shukla, K. K. (2000-01-01), Sinha, NARESH K.; Gupta, MADAN M. (eds.), "CHAPTER 17 - Soft Computing Paradigms for Artificial Vision", *Soft Computing and Intelligent Systems*, Academic Press Series in Engineering, San Diego: Academic Press, pp. 405–417, ISBN 978-0-12-646490-0, retrieved 2021-02-24

HASIL SCANNING SIMILARITY

Berisi tentang hasil scanning plagiat dengan batas toleransi 20%.

KOMENTAR REVIEWER

Berisi tentang komentar reviewer atas substansi dari buku ini sehingga mampu memperkuat bahan buku ini baik dan layak dibaca oleh para pembaca.

BIOGRAFI PENULIS

Penulis buku ini adalah Adi Widiatmoko W., S.Si., M.T., lahir di Magelang pada tanggal 5 Maret 1989 dan sejak tahun 2019 memiliki profesi sebagai dosen tetap di Program Studi D-IV Instrumentasi-MKG, Sekolah Tinggi Meteorologi Klimatologi dan Geofisika. Beberapa mata kuliah praktik yang diampu adalah Praktik Sistem Pakar, Praktik Elektronika, dan Praktik Pemrosesan Sinyal Digital. Pendidikan tinggi yang telah ditempuh penulis adalah S1 Fisika di Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Gadjah Mada, dan S2 Instrumentasi dan Kontrol di Fakultas Teknologi Industri, Institut Teknologi Bandung.

