Lampiran A ***Listing Code***

Lampiran *listing code* metode *fuzzy logic* untuk mendapatkan nilai pelepasan beban di MATLAB

```
function varargout = guifuzzy(varargin)
% GUIFUZZY MATLAB code for guifuzzy.fig
%      GUIFUZZY, by itself, creates a new GUIFUZZY or raises
the existing
%      singleton*.
%
%      H = GUIFUZZY returns the handle to a new GUIFUZZY or
the handle to
%      the existing singleton*.
%
%      GUIFUZZY('CALLBACK',hObject,eventData,handles,...)
calls the local
%      function named CALLBACK in GUIFUZZY.M with the given
input arguments.
%
%      GUIFUZZY('Property','Value',...) creates a new
GUIFUZZY or raises the
%      existing singleton*.  Starting from the left,
property value pairs are
%      applied to the GUI before guifuzzy_OpeningFcn gets
called.  An
%      unrecognized property name or invalid value makes
property application
%      stop.  All inputs are passed to guifuzzy_OpeningFcn
via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI
allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
guifuzzy

% Last Modified by GUIDE v2.5 15-Dec-2022 14:14:20

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @guifuzzy_OpeningFcn,
...
                   'gui_OutputFcn',  @guifuzzy_OutputFcn,
...
```

```
                    'gui_LayoutFcn',  [] , ...
                    'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State,
varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before guifuzzy is made visible.
function guifuzzy_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see
GUIDATA)
% varargin   command line arguments to guifuzzy (see
VARARGIN)

% Choose default command line output for guifuzzy
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);
movegui(hObject, 'center');

% UIWAIT makes guifuzzy wait for user response (see
UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command
line.
function varargout = guifuzzy_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see
VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see
GUIDATA)
```

```
% Get default command line output from handles structure
varargout{1} = handles.output;


% --- Executes on selection change in popupmenu1.
function popupmenu1_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see
GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns
popupmenu1 contents as cell array
%        contents{get(hObject,'Value')} returns selected
item from popupmenu1


% --- Executes during object creation, after setting all
properties.
function popupmenu1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: popupmenu controls usually have a white background
on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see
GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as
text
%        str2double(get(hObject,'String')) returns contents
of edit1 as a double


% --- Executes during object creation, after setting all
properties.
```

```
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see
GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as
text
%        str2double(get(hObject,'String')) returns contents
of edit2 as a double


% --- Executes during object creation, after setting all
properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
```

```
% handles    structure with handles and user data (see
GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as
text
%        str2double(get(hObject,'String')) returns contents
of edit3 as a double


% --- Executes during object creation, after setting all
properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit4_Callback(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see
GUIDATA)

% Hints: get(hObject,'String') returns contents of edit4 as
text
%        str2double(get(hObject,'String')) returns contents
of edit4 as a double


% --- Executes during object creation, after setting all
properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called
```

```
% Hint: edit controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see
GUIDATA)

data = xlsread('Book1.xlsx',2,'A3:AB12511');
%Proses Fuzzy
%input1
a= newfis('coba');
a=addvar(a,'input','beban lebih',[20 100]);
a=addmf(a,'input',1,'sk','trimf',[10 16 30]);
a=addmf(a,'input',1,'k','trimf', [30 40 50]);
a=addmf(a,'input',1,'s','trimf',[39.2 77.7 90.2]);
a=addmf(a,'input',1,'b','trimf',[54 74 90]);
a=addmf(a,'input',1,'sb','trapmf',[85 95 100 100]);
%figure; plotmf(a,'input',1)
%input2
a=addvar(a,'input','daya generator',[20 100]);
a=addmf(a,'input',2,'sk','trapmf',[0 20 25 35]);
a=addmf(a,'input',2,'k','trimf',[30 40 50]);
a=addmf(a,'input',2,'s','trimf',[40 83 90]);
a=addmf(a,'input',2,'b','trimf',[67 79 90]);
a=addmf(a,'input',2,'sb','trapmf',[85 95 100 100]);
%figure; plotmf(a,'input',2)
%output
a=addvar(a,'output','TIP',[10 110]);
a=addmf(a,'output',1,'sr','trimf',[10 20 50]);
a=addmf(a,'output',1,'r','trimf',[19 35 70]);
a=addmf(a,'output',1,'s','trimf',[35 47.1 82]);
a=addmf(a,'output',1,'t','trimf',[66 103 105]);
a=addmf(a,'output',1,'st','trapmf',[86 100 110 110]);
%figure; plotmf(a,'output',1)
%rulebase
rule1=[1 5 1 1 1];
rule2=[2 5 1 1 1];
rule3=[3 5 1 1 1];
rule4=[4 5 2 1 1];
rule5=[5 5 3 1 1];
rule6=[1 4 1 1 1];
```

```
rule7=[2 4 1 1 1];
rule8=[3 4 2 1 1];
rule9=[4 4 3 1 1];
rule10=[5 4 4 1 1];
rule11=[1 3 1 1 1];
rule12=[2 3 2 1 1];
rule13=[3 3 3 1 1];
rule14=[4 3 4 1 1];
rule15=[5 3 5 1 1];
rule16=[1 2 2 1 1];
rule17=[2 2 3 1 1];
rule18=[3 2 4 1 1];
rule19=[4 2 4 1 1];
rule20=[5 2 5 1 1];
rule21=[1 1 3 1 1];
rule22=[2 1 4 1 1];
rule23=[3 1 5 1 1];
rule24=[4 1 5 1 1];
rule25=[5 1 5 1 1];
rulelist=[rule1;rule2;rule3;rule4;rule5;...
    rule6;rule7;rule8;rule9;rule10;...
    rule11;rule12;rule13;rule14;rule15;...
    rule16;rule17;rule18;rule19;rule20;...
    rule21;rule22;rule23;rule24;rule25];
a = addrule(a,rulelist);

%menyimpan program fuzzy
%save coba.fis

%Skenario
skenario = get(handles.popupmenu1,'value');
if skenario == 1
    pg = 99.1;
    pl = 29;
    x1 = data(:,1)';
    y1 = data(:,2)';
    x2 = data(:,3)';
    y2 = data(:,4)';
elseif skenario == 2
    pg = 99;
    pl = 29.1;
    x1 = data(:,5)';
    y1 = data(:,6)';
    x2 = data(:,7)';
    y2 = data(:,8)';
elseif skenario == 3
    pg = 98.9;
    pl = 29.2;
    x1 = data(:,9)';
    y1 = data(:,10)';
    x2 = data(:,11)';
```

```
    y2 = data(:,12)';
elseif skenario == 4
    pg = 70;
    pl = 58.1;
    x1 = data(:,13)';
    y1 = data(:,14)';
    x2 = data(:,15)';
    y2 = data(:,16)';
elseif skenario == 5
    pg = 69.9;
    pl = 58.2;
    x1 = data(:,17)';
    y1 = data(:,18)';
    x2 = data(:,19)';
    y2 = data(:,20)';
elseif skenario == 6
    pg = 69.8;
    pl = 58.3;
    x1 = data(:,21)';
    y1 = data(:,22)';
    x2 = data(:,23)';
    y2 = data(:,24)';
elseif skenario == 7
    pg = 40.8;
    pl = 87.3;
    x1 = data(:,25)';
    y1 = data(:,26)';
    x2 = data(:,27)';
    y2 = data(:,28)';
end

%Menjalankan simulasi fuzzy
hasil = evalfis([pl pg],a);

%Membuat Variabel

    H=5;                                %MVA
    G=(36.5+36.375+36.25+51)/4;         %MVA
    plt=128.1;                          %MW
    tcb=0.06;                           %detik
    tr=0.05;                            %detik
    pls=0;                              %MW
    t=0;                                %detik
    f0=50;                              %Hz
    fa=50;                              %Hz
    fb=49.5;                            %Hz
    fn=50;                              %Hz
    dfdt=0;                             %Hz/detik

    t1=0;
    t2=0;
```

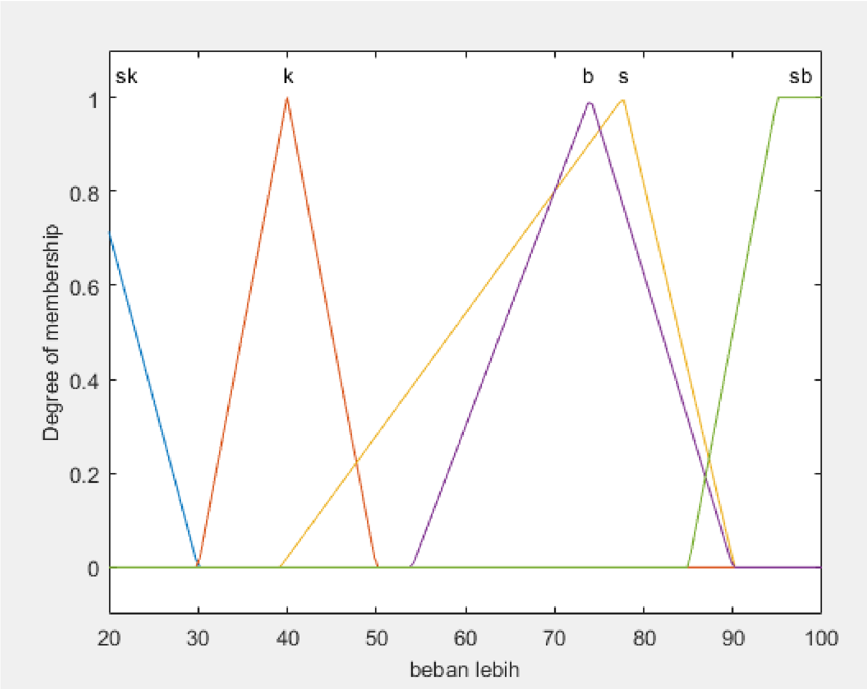
```
    t3=0;
    t4=0;
    t5=0;
    t6=0;

    %Membuat Pengulangan
    while dfdt==0
        dfdt = ((pg-(plt-pls))*f0)/(2*G*H);
        fprintf('Laju Penurunan Frekuensi = %d\n',dfdt);
        %menghitung fls
        tp = sqrt(((fa-fb)/dfdt)^2);
        ttrip= tp+tcb+tr;
        fls = fa +(dfdt*ttrip);
    end
    %kondisi awal
    told=0;
    fold=fa;
    while told < ttrip
        fnew=fold+(dfdt*told);
        told=told+0.1;

    end

    while dfdt<0
        dfdt = ((pg-(plt-pls))*f0)/(2*G*H);
        if dfdt<0
            pls = pls + hasil;
        end
    end

fold=fls;
told=ttrip;                     %waktu global
fprintf('\n');
fprintf('======= VALUE FN =======\n');
       while dfdt>0
            tnold=0;            %waktu pemulihan
          while tnold<=5
              tnnew=tnold+0.1;
              fnew=fold+dfdt*tnnew;

              fprintf('Value fn = %d\n',fnew);

              if fnew<50
                  dfdt=(fnew-fold)/(tnnew-tnold);
              else
                  fnew=50;
                  dfdt=(fnew-fold)/(tnnew-tnold);
              end

              tnew=tnnew+ttrip;
```

```
                fold=fnew;
                tnold=tnnew;

            end
        end

        if pls > 0 && pls < 30.8
           fprintf('======== TAHAP 1 ======== \n')
           tahap = 'Tahap 1';
           BebanDilepaskan = 'H13, H14, H17';
        elseif pls > 30.8 && pls < 74.4
           fprintf('======== TAHAP 2 ======== \n')
           tahap = 'Tahap 2';
           BebanDilepaskan = 'H13, H14 , H17, H11, H7, H3,
H8';
        elseif pls > 74.4 && pls < 93.4
           fprintf('======== TAHAP 3 ======== \n')
           tahap = 'Tahap 3';
           BebanDilepaskan = 'H13, H14 , H17, H11, H7, H3,
H8, H2, H4';
        end

axes(handles.axes1)
plot(x1,y1,'b-')
title(strcat('Sebelum Pelepasan Beban'))
xlabel('Waktu(s)')
ylabel('Tegangan(%)')

axes(handles.axes2)
plot(x2,y2,'g-')
title(strcat('Sesudah Pelepasan Beban'))
xlabel('Waktu(s)')
ylabel('Tegangan(%)')

pls = num2str(pls);
set(handles.edit2,'string',pls);
fls = num2str(fls);
set(handles.edit1,'string',fls);
tahap = num2str(tahap);
set(handles.edit3,'string',tahap);
BebanDilepaskan = num2str(BebanDilepaskan);
set(handles.edit4,'string',BebanDilepaskan);


% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see
GUIDATA)
```

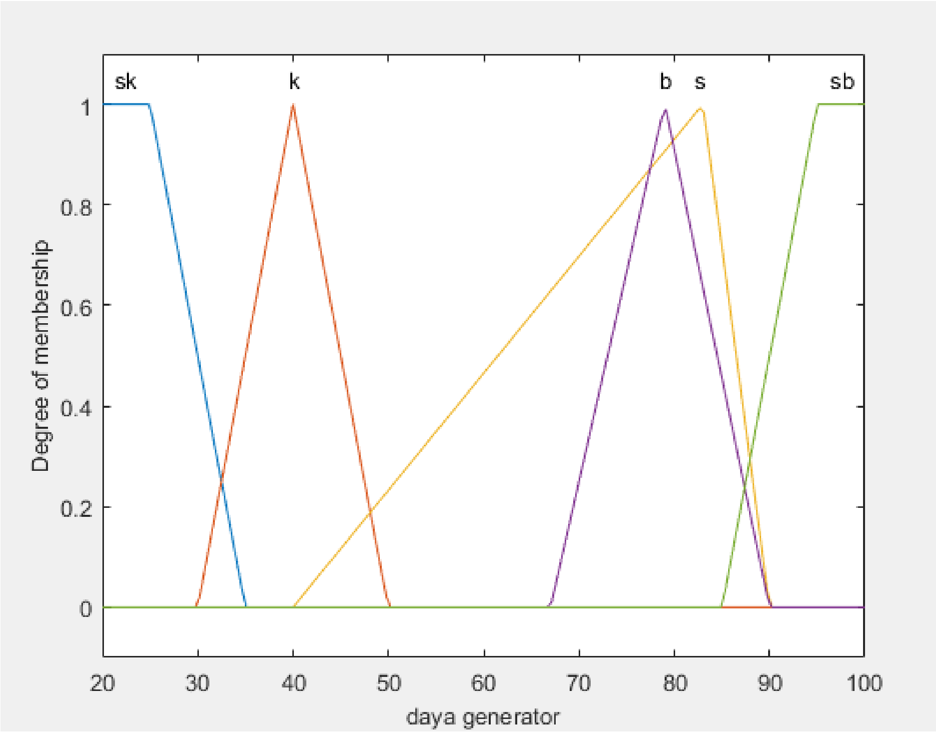
```
set(handles.edit2,'String',[])
set(handles.edit1,'String',[])
set(handles.edit3,'String',[])
set(handles.edit4,'String',[])
axes(handles.axes1)
cla reset
axes(handles.axes2)
cla reset



function edit5_Callback(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see
GUIDATA)

% Hints: get(hObject,'String') returns contents of edit5 as
text
%        str2double(get(hObject,'String')) returns contents
of edit5 as a double


% --- Executes during object creation, after setting all
properties.
function edit5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

Lampiran B *Member Function Fuzzy*

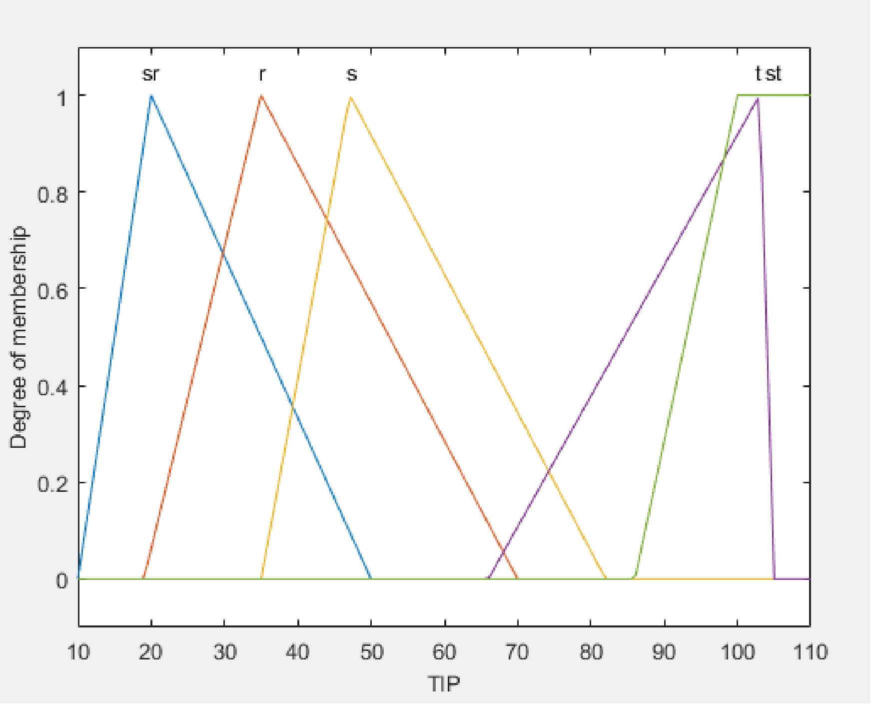
1. *Member function* beban lebih



2. *Member function* daya generator



2. *Member Function Output*



Lampiran C Data Generator

1. Data Generator Unit 1, 2, 3

TECHNICAL DATA OF GENERATOR

1. RATINGS AND SPECIFICATIONS

TYPE	OF-35-2-11
RATED OUTPUT	35 MW/43,75 MVA
RATED VOLTAGE	11 KV
RATED CURRENT	2296 A
RATED POWER FACTOR	0,80
FREQUENCY	50 HZ
SPEED	3000 r/min
NO. OF PHASES	3
STATOR WINDING CONNECTION	STAR
NO. OF TERMINALS	6
COOLING SYSTEM	CLOSED-CIRCUIT AIR COOLED
AMBIENT TEMPERATURE	5--40 DEG.C
SHORT CIRCUIT RATIO (GUAR.)	>=0.45
SUBTRANSIENT REACTANCE (GUAR.)	>=0.10
TRANSIENT REACTANCE (GUAR.)	<=0.30
EFFICIENCY (GUAR.)	>=97.5 %
BEARING PEDESTAL VIBRATION	<=0.30 mm
EXCITATION SYSTEM	STATIC THYRISTOR EXCITATION
CEILING VOLTAGE	>=1.8 p.n
VOLTAGE RESPONSE RATIO	HIGH INITIAL RESPONSE SYSTEM
DURATION FOR FORCING EXCITATION	10 S
NOISE (GUAR.)	<=90 dB (A)

2. Data Generator Unit 6

THOM ANIQUE	OPERATING AND MAINTENANCE MANUAL	Section 1 01 . 05
	DESCRIPTIVE MANUAL	Page : 01 Revision : Date :

Par.

GENERAL ELECTRICAL CHARACTERISTICS

- Cold air temperature	40°C
- Apparent power	87 500 KVA
- Power under continuous operating conditions	70 000 KW
Voltage +/- 5 %	11 000 V
- Current	4 593 A
- Cos Ø	0.8
- Frequency	50 Hz
- Speed	3 000 rpm
- Airgap	60 mm
- Air flow rate	34.2 m^3/s
- Losses to be dissipated	1 758 KW
- Number of high voltage outputs	6
- Number of phases	3
- Three-phase short circuit torque transmitted to foundations	2 600 000 mN

Lampiran D Data Transformer

SHIHLIN ELECTRIC & ENGINEERING CO.

TRANSFORMER SPECIFICATION

■ for ORDER Purpose

☐ for TENDERING Purpose

REV.: Change H.V. voltage & painting (20600V → 20600V) (19600V → 19600V) (Dark Grey → Grey Blue) Sep 15 '95

		Remark
1. Purchaser	台塑重工 (IKPP 永吉廠)	
2. Delivery Place	---	
3. Quantity Nos.	One Set.	
4. RATING: Duty / Phase / Hz / Rating KVA / H.V. Voltage / L.V. Voltage / B.I.L. (Dry) / Applied Voltage / Vector Group / Spec. Instruction	Continue Rating 3φ 50 Hz 87.5/109 MVA 21000F-20600F-20000R-19600F-19000F V 11000 V H.V. : 150 KV : L.V. : 110 KV H.V. : 50 KV : L.V. : 34 KV Ynd11 EIB-463-2A	
5. Standard	ANSI C57.12	
6. Site Conditions	Max. Ambient Temperature : 40 ℃ Altitude : 1000 M Below Service Location : Out Door	
7. Type	Oil Immersed	
8. Cooling Type	OA/FA	
9. Oil presevation	Breather with Conservator.	
10. External connect -tion Method	H.V. Side : Vertical L.V. Side : Vertical	
11. Characteristics (Rated Voltage & Frequency at OA Load, 75℃ P.F=1.0)	No-load : ----- W Total Loss: 439700 W Efficity : 99.50 % % IX : 11.0 % Excitation Current : 1.2 % Voltage Regulation(%): 1.3 % Audible Level : 80 dB	
12. Temperature Rise	Winding : 65 ℃ : Oil : 65 ℃	
13. Approximate Weight & Dimension	Length :(L) ---- mm Width :(W) ---- mm Height :(H) ---- mm Oil : 28000 Litre Total Weight: 115000 Kg	
14. Outline Drawing	E 314932	
15. Painting Color	No. 1-94 (Grey Blue) (灰藍)	
16. Special condition	-----	

Prepared: C T. Shiau Sep 15 '95 Checked/Approved: Id c Shiu Sep 15 '95 Spec. No.:SH03500445(1/2)

(lanjutan)

TRANSFORMATOR OIL IMMERSED PT DSS

NO	NAME	RATED POWER	TYPE OF COOLING	RATED VOLTAGE	RATED CURRENT	% Z	PHASA	F	VECTOR GROUP
1	*Main* Transformator unit #1, #2, dan #3	30,5/44 MVA	ONAN/ ONAF	HV 20000 / LV 11000 V	HV 1270.2/ LV 2309.5 A	10.50%	3 ϕ	50 Hz	YNd11
2	*Main* Transformator Unit #6	87,5/109 MVA	OA/FA	LV 11000/ HV 20000 V	LV 4593/5721 / HV 2526/3147 A	10.94%	3 ϕ	50 Hz	YNd11
3	HV *Stand By* Transformato #2	7000 / 10000 KVA	ONAN /ONAF	HV 20 ± 3 x 2.5% / LV 3,45 KV	HV : 524.9 / LV : 1073.5 A	7.23%	3 ϕ	50 Hz	YN,d11
4	HV *Stand By* Transformato #1	7000/10000 KVA	ONAN/ ONAF	HV 20 +- 3x2.5% /LV 3.45 KV	HV : 288.7 / LV : 1673.5 A	7.62%	3 ϕ	50 Hz	Ynd11
5	HV Auxillary Unit #1	10000 KVA	ONAN/ ONAF	HV 11000 / LV 3450 V	HV 524.9 / LV 1673.5 A	7.43%	3 ϕ	50 Hz	D.d0
6	HV Auxillary Unit #2	10000 KVA	ONAN/ ONAF	HV 11000 / LV 3450 V	HV 524.9 / LV 1673.5 A	7.43%	3 ϕ	50 Hz	D.d0
7	HV Auxillary Unit #3	10000 KVA	ONAN/ ONAF	HV 11000 / LV 3450 V	HV 524.9 / LV 1673.5 A	7.43%	3 ϕ	50 Hz	D.d0
8	HV Auxillary Unit #6	12500/15600 KVA	OA/FA	HV 11000/ LV 3450 V	HV 656 / LV 818.78 A	6.50%	3 ϕ	50 Hz	Dd0
9	TRANSFORMATOR DIESEL 1	5000 KVA	ONAN	HV 11000/ LV 20000 KV	LV 262,4/ HV 144,3 A	6.70%	3 ϕ	50 Hz	Dyn11
10	TRANSFORMATOR DIESEL 2	5000 KVA	ONAN	HV 11000/ LV 20000 KV	LV 262,4/ HV 144,3 A	6.70%	3 ϕ	50 Hz	Dyn11
11	TRANSFORMATOR DIESEL 3	5000 KVA	ONAN	HV 11000/ LV 20000 KV	HV 144.3 A / LV 262.4 A		3 ϕ	50 Hz	YNd11
12	TRANSFORMATOR DIESEL 4	5000 KVA	ONAN	HV 11000/ LV 20000 KV	HV 144.3 A / LV 262.4 A		3 ϕ	50 Hz	YNd11
13	TRANSFORMATOR DIESEL 5	4000/5000 KVA	ONAN /ONAF	HV 11000/ LV 20000 KV	HV 115,47/144,34 / LV 209.94/262,43	6.05%	3 ϕ	50 Hz	YNd11

(lanjutan)

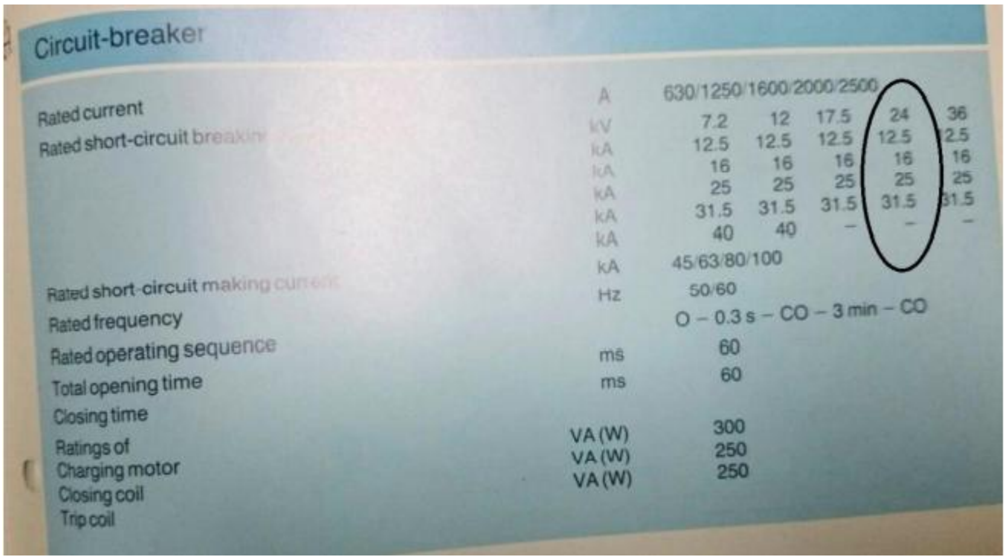
14	STATIONARY TRANSFORMATOR W/T	2500 KVA	ONAN	HV 20000/ LV 400 KV	PRIMER 72,17/SECONDARY 2508 A	6.50%	3 ϕ	50 Hz	Dyn 5
15	TRANSFORMATOR MCC 1 W/T	4000/5000 KVA	OA/FA	HV 20000/ LV 11000 KV	HV OA:115A FA:134A / LV OA: 210A FA : 262 A	6,0 %	3 ϕ	50Hz	Dyn11
16	TRANSFORMATOR MCC 2 W/T	1500 KVA	OA	HV 3300 / LV 380/220 V	HV 262 / LV 2279 A	6,1 %	3 ϕ	50Hz	Dy11
17	NO 15 W/T	1250/1400-1438/1610 KVA	OA/FFA	HV 3300/ LV 380		6,12 %	3 ϕ	50 Hz	DnY
18	TRANSFORMATOR #1 #2 #3 #6 CHS	800 KVA	ONAN	HV 3.45+-5% / LV 0.4 KV	HV 133.9/ LV 1154.7 A	8%	3 ϕ	50 Hz	D, yn11
19		10000 KVA	ONAN/ ONAF	HV 20000/ LV 3450 V	HV 288.7 / LV 1673.5 A	7.23%	3 ϕ	50 Hz	YN, dll
20	TRANSFORMATOR PLN 150 KV	10000/12500 KVA	ONAN/ ONAF	HV 20000 (8+- x1.25%)/ LV 3450 V	HV 0.288 / LV 2.09 A	7.23%	3 ϕ	50 Hz	YNd11

Mengetahui
Kepala Divisi Electical & Instrumen

PPIC 04 JUN 2018 SERANG

Lili Sugeng Diko Nugroho
NIP : 983312

Lampiran E Data *Circuit Breaker*

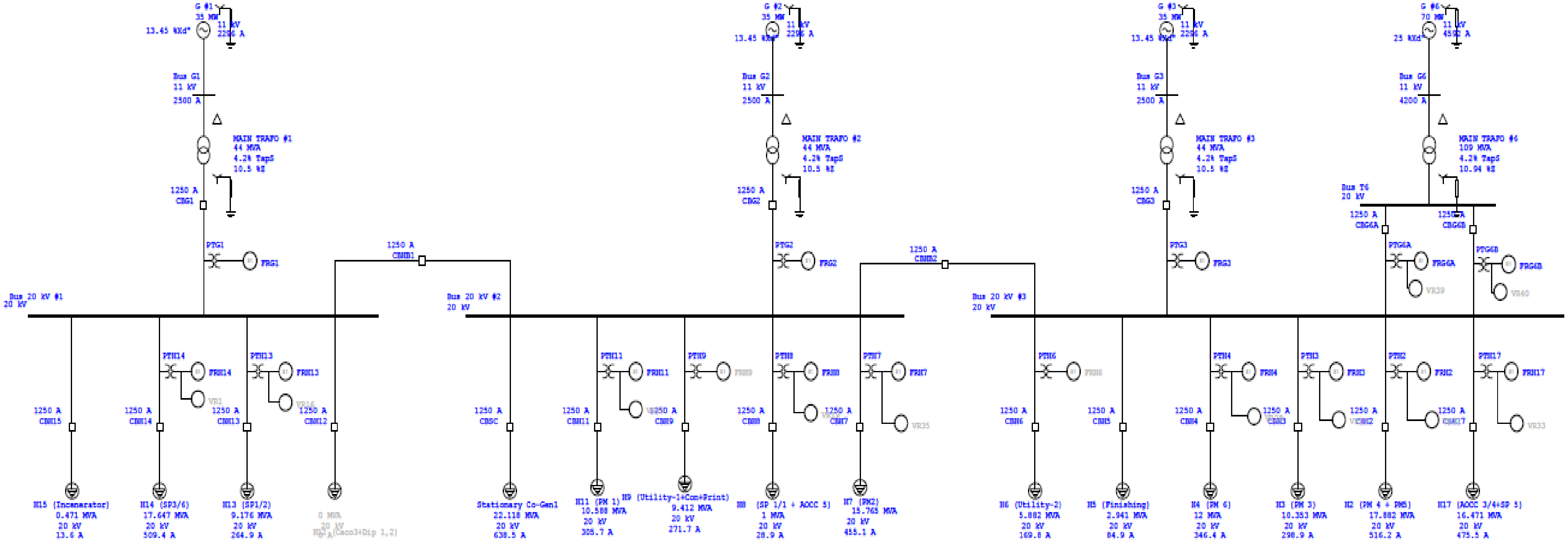


Circuit-breaker

Rated current
Rated short-circuit breakin

	A	630/1250/1600/2000/2500				
	kV	7.2	12	17.5	24	36
	kA	12.5	12.5	12.5	12.5	12.5
	kA	16	16	16	16	16
	kA	25	25	25	25	25
	kA	31.5	31.5	31.5	31.5	31.5
	kA	40	40	–	–	–
Rated short-circuit making curre	kA	45/63/80/100				
Rated frequency	Hz	50/60				
Rated operating sequence		O – 0.3 s – CO – 3 min – CO				
Total opening time	ms	60				
Closing time	ms	60				
Ratings of						
Charging motor	VA (W)	300				
Closing coil	VA (W)	250				
Trip coil	VA (W)	250				

Lampiran F *Single Line* Sistem pada ETAP



Lampiran G Perhitungan Laju Penurunan Frekuensi

No	GENERATOR				Jumlah Suplai (MW)	Jumlah Beban (MW)	Beban Lebih (MW)	df/dt
	G#1	G#2	G#3	G#6				
1	29,2	29,1	29	40,8	99,1	128,1	29	-3,622
2	29,2	29,1	29	40,8	99	128,1	29,1	-3,635
3	29,2	29,1	29	40,8	98,9	128,1	29,2	-3,647
4	29,2	29,1	29	40,8	87,3	128,1	40,8	-5,096
5	29,2	29,1	29	40,8	70	128,1	58,1	-7,257
6	29,2	29,1	29	40,8	69,9	128,1	58,2	-7,269
7	29,2	29,1	29	40,8	69,8	128,1	58,3	-7,282
8	29,2	29,1	29	40,8	58,3	128,1	69,8	-8,718
9	29,2	29,1	29	40,8	58,2	128,1	69,9	-8,731
10	29,2	29,1	29	40,8	58,1	128,1	70	-8,743
11	29,2	29,1	29	40,8	40,8	128,1	87,3	-10,904
12	29,2	29,1	29	40,8	29,2	128,1	98,9	-12,353
13	29,2	29,1	29	40,8	29,1	128,1	99	-12,365
14	29,2	29,1	29	40,8	29	128,1	99,1	-12,378

Keterangan Warna:

= Generator Padam

= Generator Normal

Perhitungan Waktu *Trip* Masing-Masing Skenario Generator Padam

No	GENERATOR				df/dt	f_0	f_1	$t_{pick\ up}$ (s)	t_{cb} (s)	t_{relay} (s)	t_{trip} (s)
	G #1	G #2	G #3	G #6							
1	29,2	29,1	29	40,8	-3,622	50	49,5	0,138	0,06	0,05	0,248
2	29,2	29,1	29	40,8	-3,635	50	49,5	0,138	0,06	0,05	0,248
3	29,2	29,1	29	40,8	-3,647	50	49,5	0,137	0,06	0,05	0,247
4	29,2	29,1	29	40,8	-5,096	50	49,5	0,098	0,06	0,05	0,208
5	29,2	29,1	29	40,8	-7,257	50	49,5	0,069	0,06	0,05	0,179
6	29,2	29,1	29	40,8	-7,269	50	49,5	0,069	0,06	0,05	0,179
7	29,2	29,1	29	40,8	-7,282	50	49,5	0,069	0,06	0,05	0,179
8	29,2	29,1	29	40,8	-8,718	50	49,5	0,057	0,06	0,05	0,167
9	29,2	29,1	29	40,8	-8,731	50	49,5	0,057	0,06	0,05	0,167
10	29,2	29,1	29	40,8	-8,743	50	49,5	0,057	0,06	0,05	0,167
11	29,2	29,1	29	40,8	-10,904	50	49,5	0,046	0,06	0,05	0,156
12	29,2	29,1	29	40,8	-12,353	50	49,5	0,040	0,06	0,05	0,150
13	29,2	29,1	29	40,8	-12,365	50	49,5	0,040	0,06	0,05	0,150
14	29,2	29,1	29	40,8	-12,378	50	49,5	0,040	0,06	0,05	0,150

Keterangan Warna:

= Generator Padam

= Generator Normal

Perhitungan Frekuensi Saat *Circuit Breaker* Bekerja Pada Masing-Masing Skenario Generator Padam

No	GENERATOR				df/dt	T_{trip} (s)	F_{LS} (Hz)
	G #1	G #2	G #3	G #6			
1	29,2	29,1	29	40,8	-3,622	0,248	49,102
2	29,2	29,1	29	40,8	-3,635	0,248	49,100
3	29,2	29,1	29	40,8	-3,647	0,247	49,099
4	29,2	29,1	29	40,8	-5,096	0,208	48,939
5	29,2	29,1	29	40,8	-7,257	0,179	48,702
6	29,2	29,1	29	40,8	-7,269	0,179	48,700
7	29,2	29,1	29	40,8	-7,282	0,179	48,699
8	29,2	29,1	29	40,8	-8,718	0,167	48,541
9	29,2	29,1	29	40,8	-8,731	0,167	48,540
10	29,2	29,1	29	40,8	-8,743	0,167	48,538
11	29,2	29,1	29	40,8	-10,904	0,156	48,301
12	29,2	29,1	29	40,8	-12,353	0,150	48,141
13	29,2	29,1	29	40,8	-12,365	0,150	48,140
14	29,2	29,1	29	40,8	-12,378	0,150	48,138

Keterangan Warna:

= Generator Padam

= Generator Normal

Perhitungan Laju Pemulihan Frekuensi yang Harus Dihasilkan dengan Waktu 5 Detik dari Setiap Kombinasi Generator Padam

No	GENERATOR				F_{LS}	Δf	df/dt pemulihan
	G #1	G #2	G #3	G #6			
1	29,2	29,1	29	40,8	49,102	0,89843872	0,1797
2	29,2	29,1	29	40,8	49,100	0,899812646	0,1800
3	29,2	29,1	29	40,8	49,099	0,901186573	0,1802
4	29,2	29,1	29	40,8	48,939	1,060562061	0,2121
5	29,2	29,1	29	40,8	48,702	1,298251366	0,2597
6	29,2	29,1	29	40,8	48,700	1,299625293	0,2599
7	29,2	29,1	29	40,8	48,699	1,300999219	0,2602
8	29,2	29,1	29	40,8	48,541	1,459000781	0,2918
9	29,2	29,1	29	40,8	48,540	1,460374707	0,2921
10	29,2	29,1	29	40,8	48,538	1,461748634	0,2923
11	29,2	29,1	29	40,8	48,301	1,699437939	0,3399
12	29,2	29,1	29	40,8	48,141	1,858813427	0,3718
13	29,2	29,1	29	40,8	48,140	1,860187354	0,3720
14	29,2	29,1	29	40,8	48,138	1,86156128	0,3723

Keterangan Warna:

= Generator Padam

= Generator Normal

Perhitungan Beban Lepas Pada Setiap Kombinasi Generator Padam

No	GENERATOR				df/dt pemulihan	Beban yang dilepas (MW)
	G #1	G #2	G #3	G #6		
1	29,2	29,1	29	40,8	0,1797	29,77307187
2	29,2	29,1	29	40,8	0,1800	29,87427575
3	29,2	29,1	29	40,8	0,1802	29,97547969
4	29,2	29,1	29	40,8	0,2121	41,71559587
5	29,2	29,1	29	40,8	0,2597	59,22626593
6	29,2	29,1	29	40,8	0,2599	59,32748965
7	29,2	29,1	29	40,8	0,2602	59,42871345
8	29,2	29,1	29	40,8	0,2918	71,0699117
9	29,2	29,1	29	40,8	0,2921	71,17114355
10	29,2	29,1	29	40,8	0,2923	71,27237546
11	29,2	29,1	29	40,8	0,3399	88,78655108
12	29,2	29,1	29	40,8	0,3718	100,5313447
13	29,2	29,1	29	40,8	0,3720	100,632597
14	29,2	29,1	29	40,8	0,3723	100,7338495

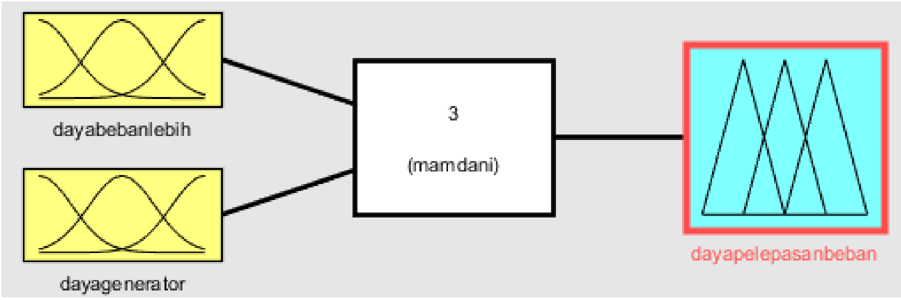
Keterangan Warna:

= Generator Padam

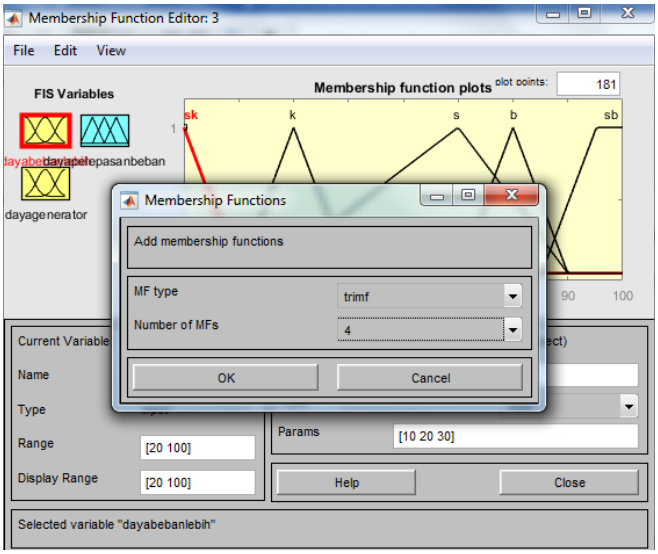
= Generator Normal

Lampiran H *Desain Fuzzy Logic*

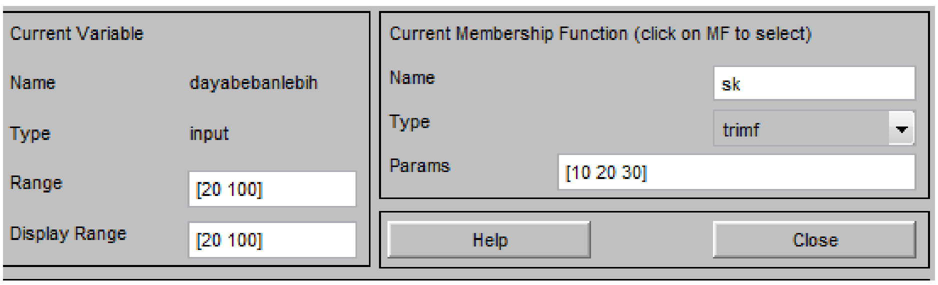
I. Blok Variabel *Input* dan *Output* pada *Fuzzy logic Designer*



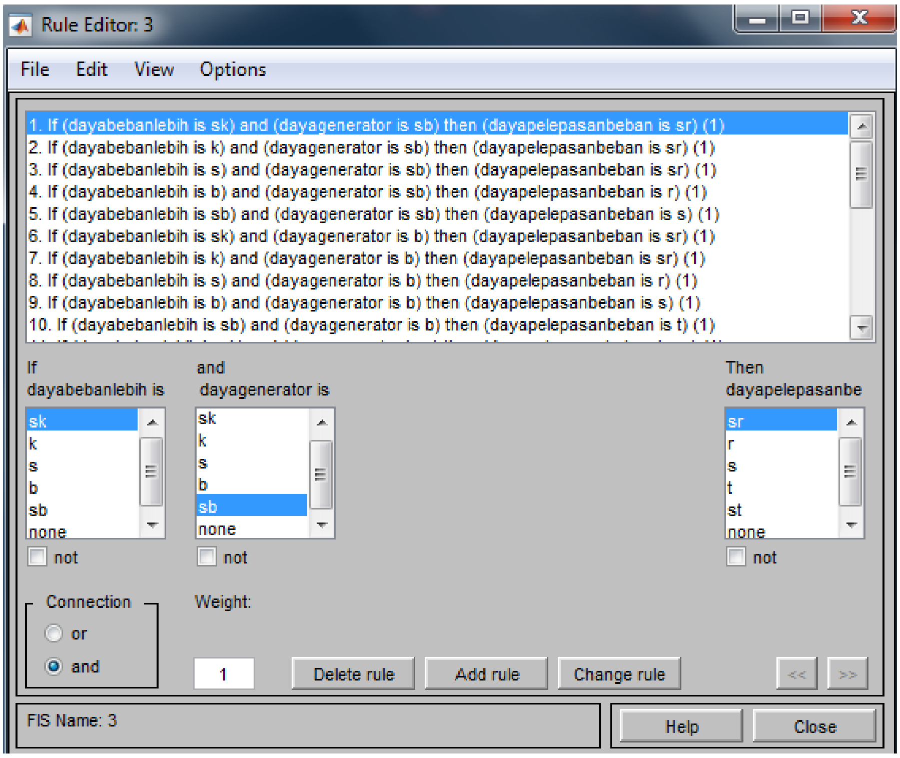
II. *Editor* Fungsi Keanggotaan



III. Kolom *Range*, *Params*, dan *Name* pada *Editor* Fungsi Keanggotaan



IV. Tampilan *Rule Editor*



V. *Input* dan Hasil pada *Rule Viewer*