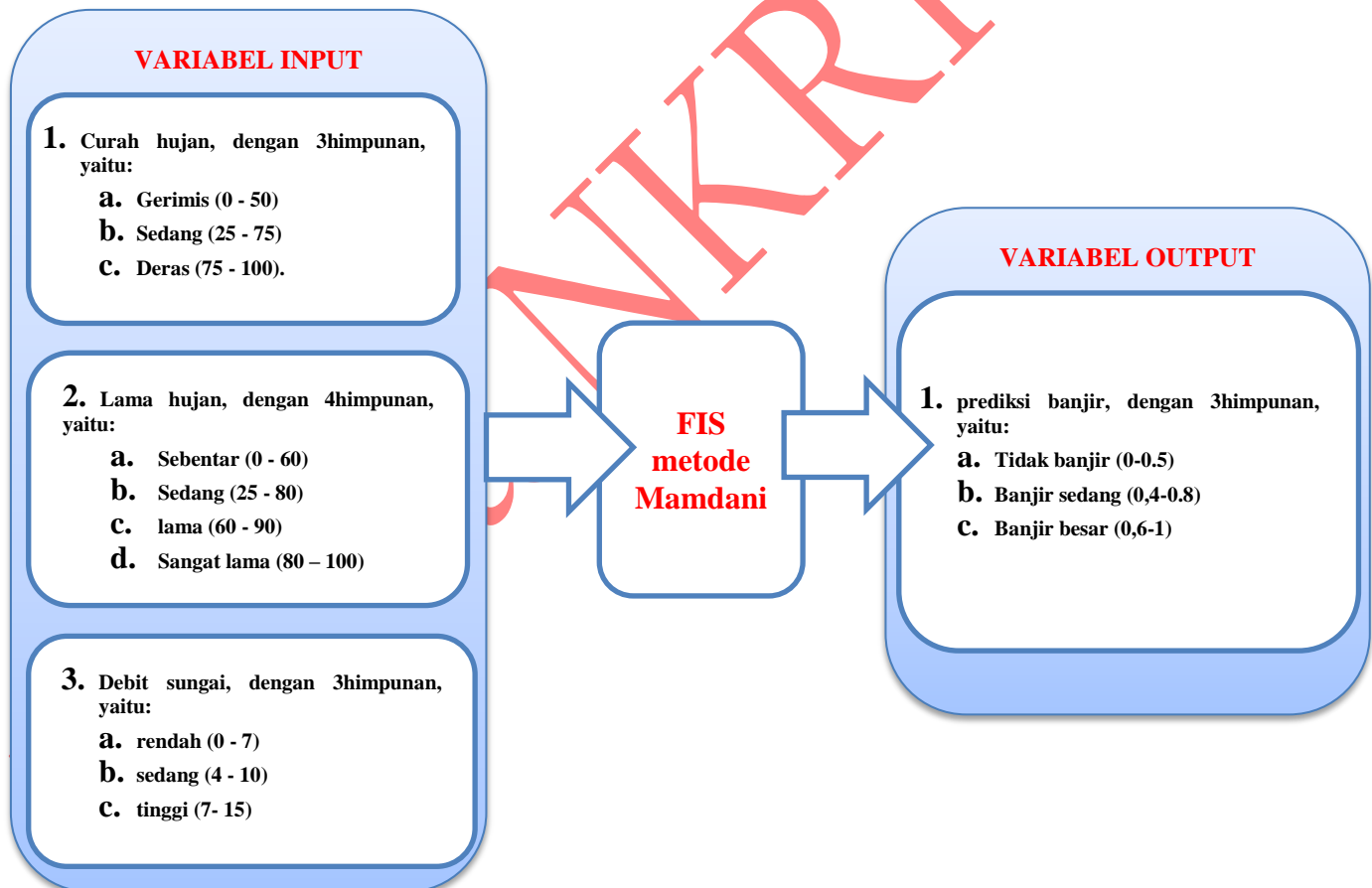


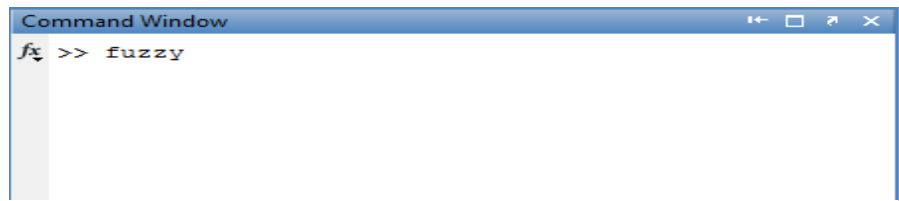
# Penggunaan Matlab Untuk Perhitungan dan Simulasi Sistem Prediksi Banjir Menggunakan Fuzzy Inference System Metode Mamdani

Hal terpenting dalam membangun sistem fuzzy adalah seberapa jauh seorang pembuat sistem memahami dan mengerti tentang sistem yang akan ia buat ( berkenaan dengan parameter – parameter yang berpengaruh terhadap sistem ataupun berkenaan dengan masalah – masalah dari sebuah kasus yang ingin dibuatkan sistem fuzzy). Dalam kasus ini, untuk memprediksi datangnya banjir setidaknya dipengaruhi oleh beberapa parameter, yaitu: curah hujan, lamanya hujan dan debit air di sungai. Sehingga kerangka pemikiran yang digunakan untuk membangun sistem prediksi banjir dengan menggunakan FIS metode mamdani dapat dilihat pada gambar dibawah ini:

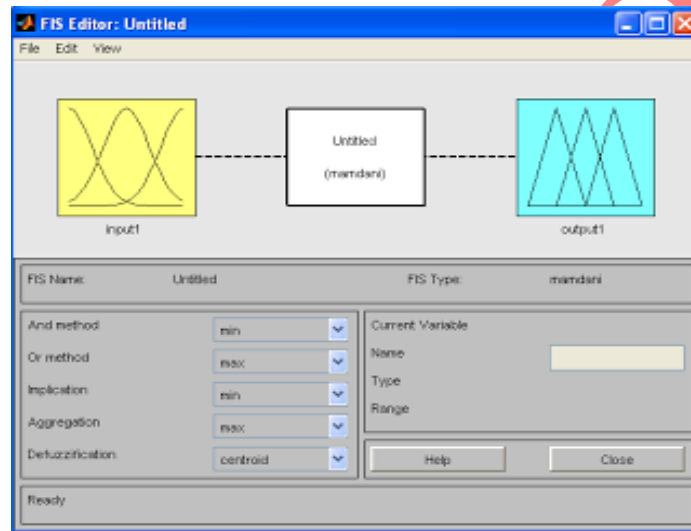


Setelah kerangka pemikiran dari sistem yang akan dibangun telah terbentuk, langkah selanjutnya adalah melakukan perhitungan dan simulasi menggunakan software matlab, berikut adalah langkah – langkah pengerjaanya:

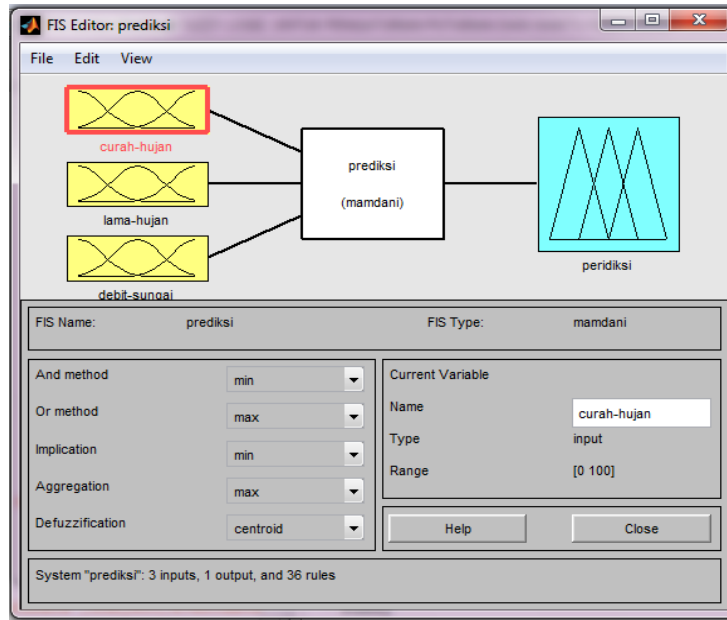
1. Jalankanlah program matlab anda lalu kemudian untuk memulai perhitungan dan simulasi ketik "fuzzy" di command window.



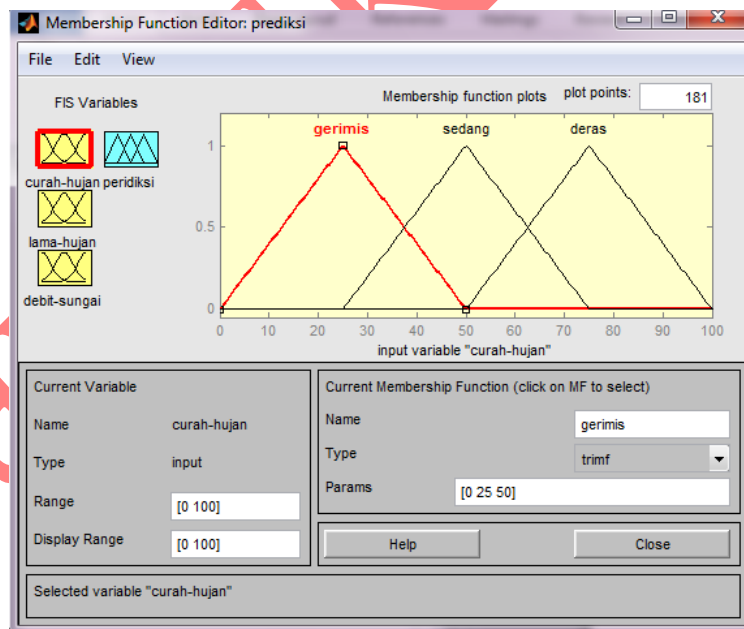
Sehingga muncul tampilan sebagai berikut.



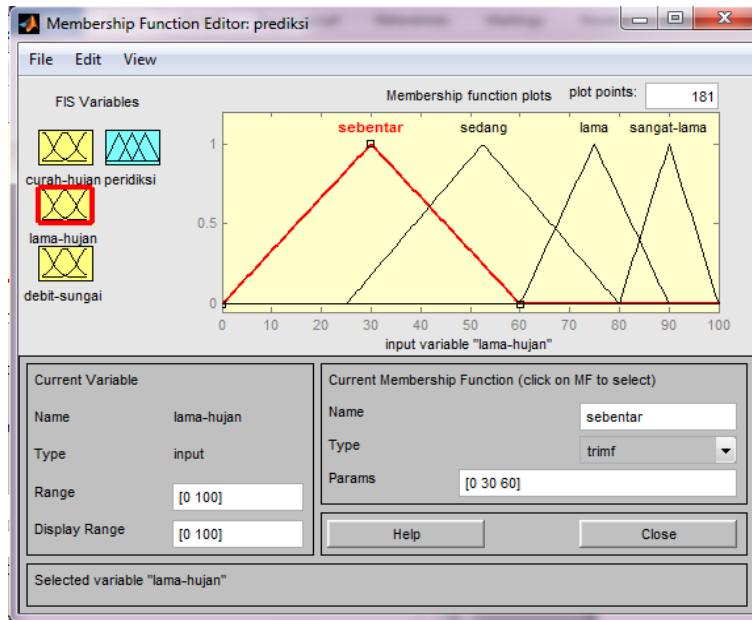
2. Klik edit-add variable-input untuk menentukan banyaknya input yang kita inginkan. Dan klik edit-add variable-output untuk menentukan output yang kita inginkan seperti tampilan dibawah ini



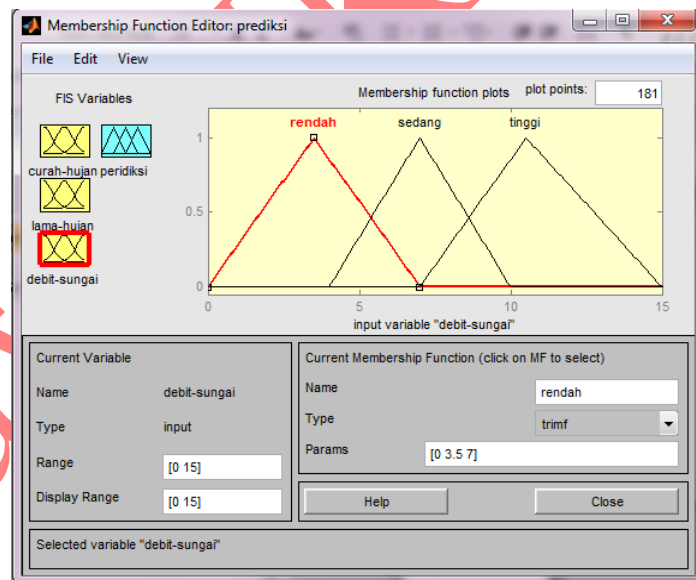
3. Klik curah hujan 2 kali. kemudian masukan himpunan fungsi keanggotaan berdasarkan data di atas. untuk kurva segitiga gunakan trimf dengan semesta pembicaraan (range) 0-100. Untuk menambahkan garis kurva di membership function editor. klik edit-MFs. Sehingga diperoleh tampilan sebagai berikut :



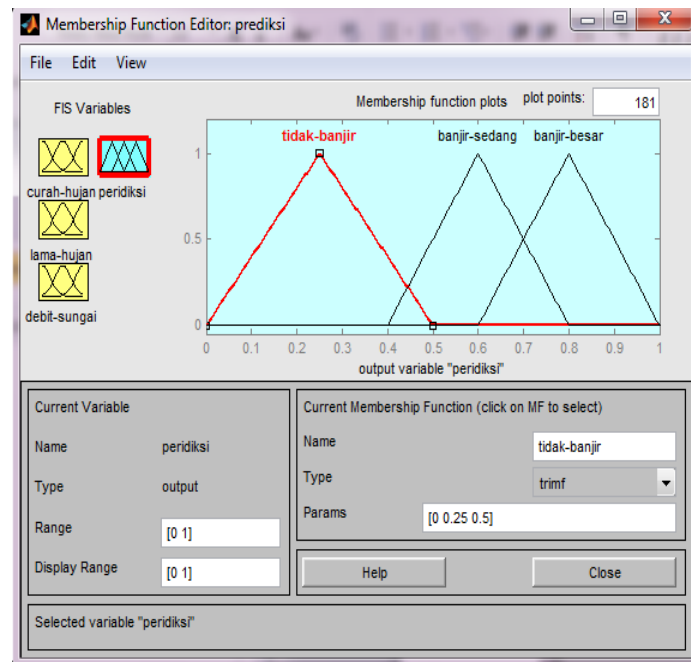
4. Klik lama hujan 2 kali. kemudian masukan himpunan fungsi keanggotaan berdasarkan data di atas. untuk kurva segitiga gunakan trimf dengan semesta pembicaraan (range) 0-100. Untuk menambahkan inputan di membership function editor. klik edit-MFs. Sehingga diperoleh tampilan sebagai berikut :



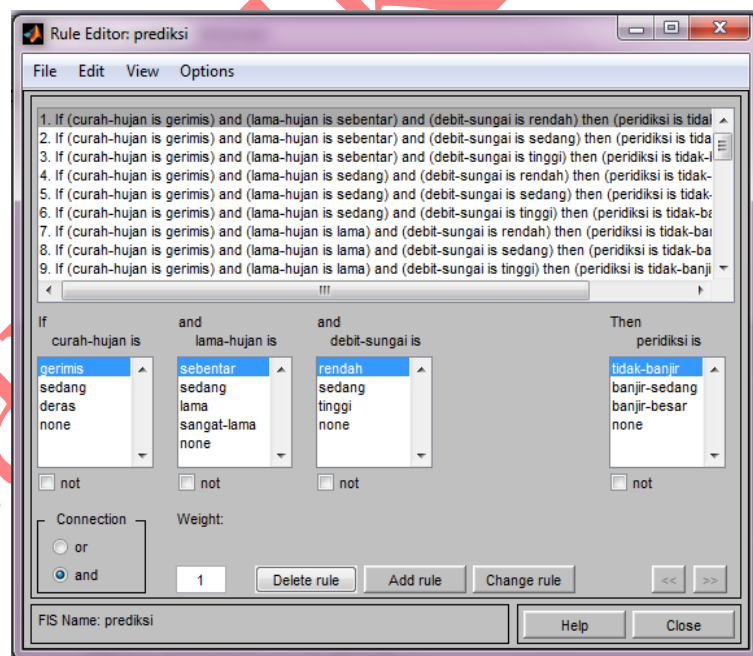
5. Klik debit sungai 2 kali. kemudian masukan himpunan fungsi keanggotaan berdasarkan data di atas. untuk segitiga gunakan trimf dengan semesta pembicaraan (range) 0-15. Untuk menambahkan inputan di membership function editor. klik edit-MFs. Sehingga diperoleh tampilan sebagai berikut :



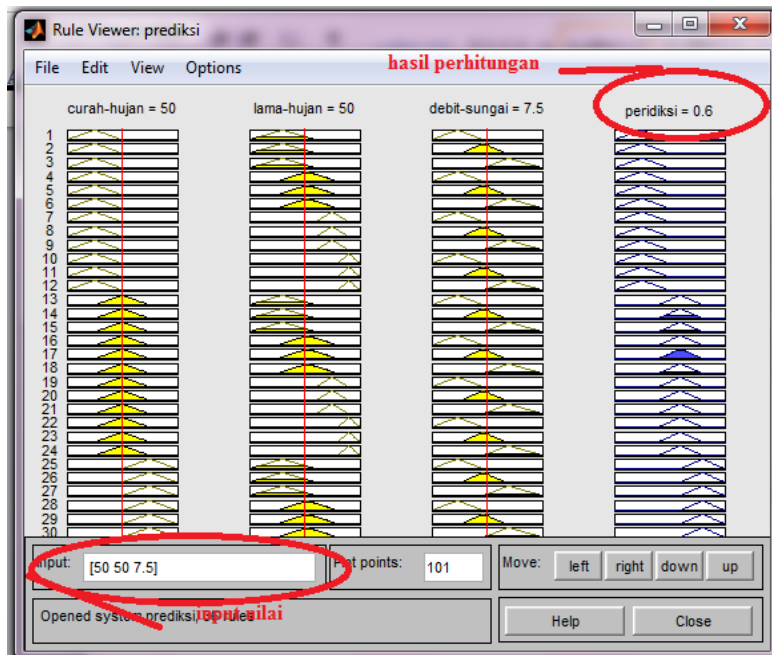
6. Klik prediksi banjir 2 kali. kemudian masukan himpunan fungsi keanggotaan berdasarkan data di atas. untuk segitiga gunakan trimf dengan semesta pembicaraan (range) 0-1. Untuk menambahkan inputan di membership function editor. klik edit-MFs. Sehingga diperoleh tampilan sebagai berikut :



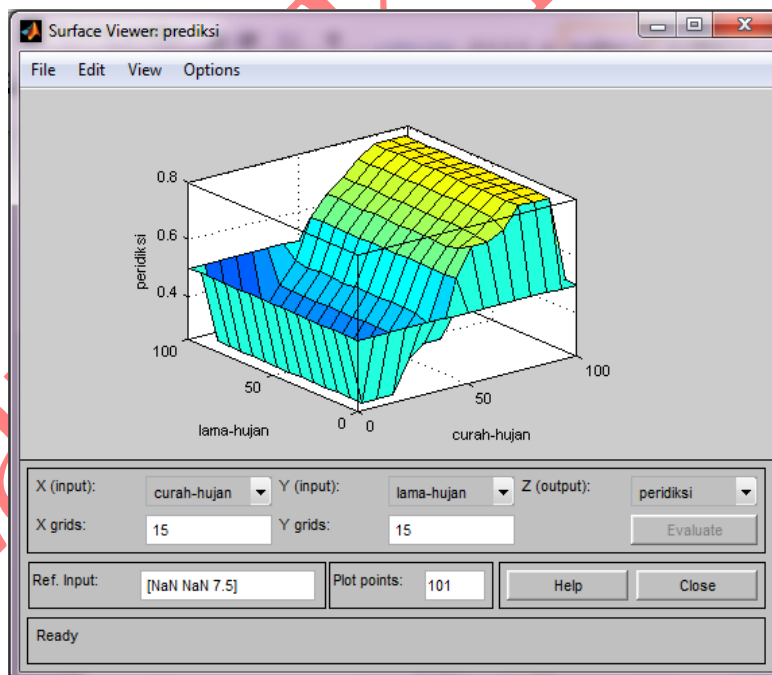
7. Setelah selesai memasukan variabel input dan output, tahap selanjutnya adalah membuat aturan-aturan berdasarkan basis pengetahuan. Klik edit-Rules dan tulis aturan berdasarkan input dan output yang kita buat. Maka diperoleh tampilan sebagai berikut.



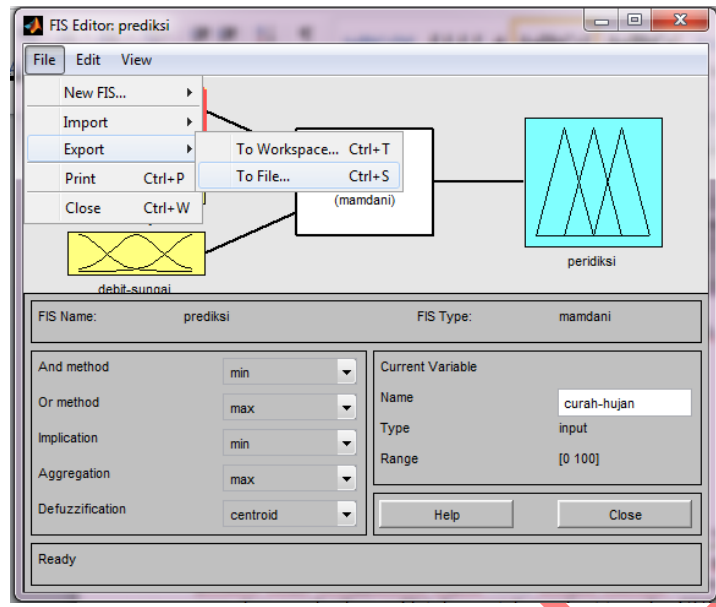
8. Setelah aturan dibuat. kemudian kita klik pada Rule-editor yaitu view-Rules maka muncul tampilan dibawah ini, yang digunakan untuk melakukan perhitungan dan simulasi sesuai dengan rule yang telah diberikan



9. Jika ingin melihat grafik maka klik view-surface pada tampilan Rule Viewer. Untuk menyimpan file tersebut.

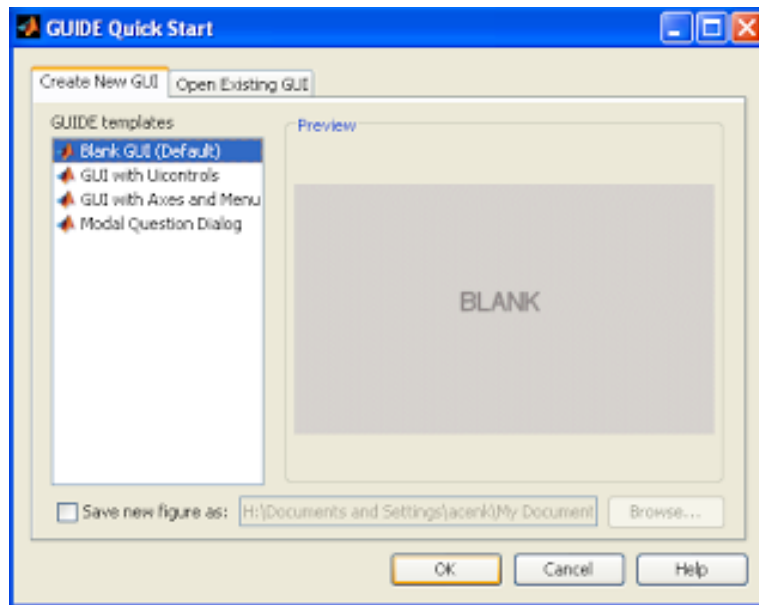


10. Langkah terakhir adalah menyimpan file sistem fuzzy yang telah anda buat, klik File-Export-To file Seperti tampilan dibawah ini.



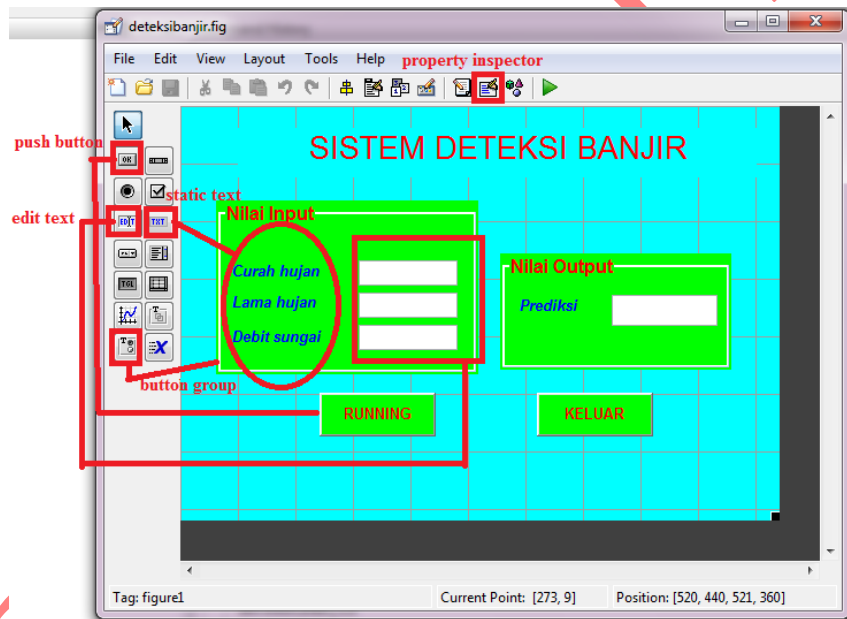
## Aplikasi prediksi banjir berbasis GUI

Setelah selesai membuat sistem fuzzy untuk perhitungan dan simulasi, pada program matlab dapat juga digunakan untuk membuat aplikasi prediksi banjir sederhana. Hal penting yang harus anda ingat adalah bahwa kita harus menyimpan file prediksi banjir yang telah diselesaikan dengan logika fuzzy dan file GUI untuk menyelesaikannya dalam satu folder, setelah itu anda masuk kedalam proses perancangan aplikasi prediksi banjir sederhana. Ketik guide pada command windows sehingga muncul tampilan sebagai berikut:

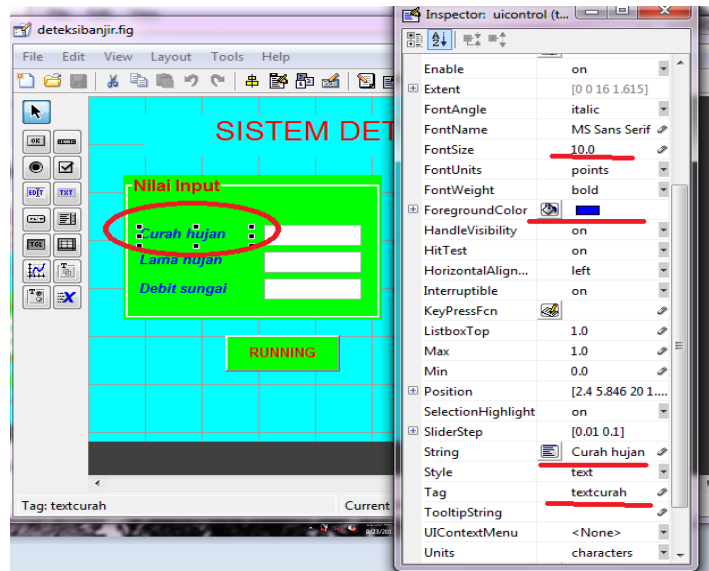


Kemudian desainlah form pada jendela GUI tersebut, dengan menggunakan 2 buah Button Group, 2buah push button, 4buah edit text dan 5buah static text. Sehingga tampilanya seperti gambar dibawah ini:





Untuk melakukan pengaturan terhadap tiap – tiap komponen dapat dilakukan dengan cara mengklik tombol property inspector atau bisa juga dengan cara double klik komponen yang bersangkutan.



Setelah desain form selesai dibuat cobalah anda run, kemudian tahap selanjutnya adalah memasukkan kode program (source code) berikut kedalam jendela editor, klik view/editor, sehingga muncul jendela editor, seperti dibawah ini.

```

1 function Increment value near cursor and evaluate cell (Ctrl+NumPad +)
2 % DETEKSIBANJIR MATLAB code for deteksibanjir.fig
3 %
4 % DETEKSIBANJIR, by itself, creates a new DETEKSIBANJIR or raises the existing
5 % singleton*.
6 %
7 % H = DETEKSIBANJIR returns the handle to a new DETEKSIBANJIR or the handle to
8 % the existing singleton*.
9 %
10 % DETEKSIBANJIR('CALLBACK',hObject,eventData,handles,...) calls the local
11 % function named CALLBACK in DETEKSIBANJIR.M with the given input arguments.
12 %
13 % DETEKSIBANJIR('Property','Value',...) creates a new DETEKSIBANJIR or raises th
14 % existing singleton*. Starting from the left, property value pairs are
15 % applied to the GUI before deteksibanjir_OpeningFcn gets called. An
16 % unrecognized property name or invalid value makes property application
17 % stop. All inputs are passed to deteksibanjir_OpeningFcn via varargin.
18 %
19 % *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
20 % instance to run (singleton)".
21 %
22 % See also: GUIDE, GUIDATA, GUIHANDLES
23 %
24 % Edit the above text to modify the response to help deteksibanjir
25 % Last Modified by GUIDE v2.5 23-Sep-2013 01:25:34

```

*\*\*hanya code yang bergaris bawah dan berwarna merah yang anda masukan dalam jendela editor untuk memanggil fungsi dari tiap – tiap komponen yang bersangkutan..!*

```

function varargout = deteksibanjir(varargin)
% DETEKSIBANJIR MATLAB code for deteksibanjir.fig
% DETEKSIBANJIR, by itself, creates a new DETEKSIBANJIR or raises the existing
% singleton*.
%
% H = DETEKSIBANJIR returns the handle to a new DETEKSIBANJIR or the handle to
% the existing singleton*.
%
% DETEKSIBANJIR('CALLBACK',hObject,eventData,handles,...) calls the local

```

```

%      function named CALLBACK in DETEKSIBANJIR.M with the given input arguments.
%
%      DETEKSIBANJIR('Property','Value',...) creates a new DETEKSIBANJIR or raises the
%      existing singleton*. Starting from the left, property value pairs are
%      applied to the GUI before deteksibanjir_OpeningFcn gets called. An
%      unrecognized property name or invalid value makes property application
%      stop. All inputs are passed to deteksibanjir_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 deteksibanjir

% Last Modified by GUIDE v2.5 22-Sep-2013 13:02:35

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @deteksibanjir_OpeningFcn, ...
                  'gui_OutputFcn',  @deteksibanjir_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 deteksibanjir is made visible.
function deteksibanjir_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 deteksibanjir (see VARARGIN)

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

% Update handles structure
guidata(hObject, handles);

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

% --- Outputs from this function are returned to the command line.
function varargout = deteksibanjir_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 button press in run.
function run_Callback(hObject, eventdata, handles)
% hObject    handle to run (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
a=readfis('prediksi')
out=evalfis(handles.Chujan handles.lamahuj handles.debit,a)

```

```
set(handles.prediksi,'string',out);
```

```
% --- Executes on button press in exit.
```

```
function exit_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to exit (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
```

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

```
close
```

```
function prediksi_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to prediksi (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 prediksi as text
```

```
%          str2double(get(hObject,'String')) returns contents of prediksi as a double
```

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

```
function prediksi_CreateFcn(hObject, eventdata, handles)
```

```
% hObject    handle to prediksi (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 Chujan_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to Chujan (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 Chujan as text
```

```
%          str2double(get(hObject,'String')) returns contents of Chujan as a double
```

```
Chujan=str2double(get(hObject,'string'));
```

```
handles.curah_hujan=Chujan;
```

```
guidata(hObject, handles);
```

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

```
function Chujan_CreateFcn(hObject, eventdata, handles)
```

```
% hObject    handle to Chujan (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 lamahuj_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to lamahuj (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 lamahuj as text
```

```
%          str2double(get(hObject,'String')) returns contents of lamahuj as a double
```

```
lamahuj=str2double(get(hObject,'string'));
```

```
handles.lamahuj=lamahuj;
```

```
guidata(hObject, handles);
```

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

```
function lamahuj_CreateFcn(hObject, eventdata, handles)
```

```
% hObject    handle to lamahuj (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 debit_Callback(hObject, eventdata, handles)
% hObject    handle to debit (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 debit as text
% str2double(get(hObject,'String')) returns contents of debit as a double

debit=str2double(get(hObject,'string'));
handles.debit=debit;
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function debit_CreateFcn(hObject, eventdata, handles)
% hObject    handle to debit (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

```

setelah itu cobalah anda run untuk mengetahui apakah terjadi kesalahan kode program atau tidak, jika tidak terjadi kesalahan maka tampilan aplikasi akan seperti dibawah ini:

deteksi banjir

## SISTEM DETEKSI BANJIR

**Nilai Input**

Curah hujan	30
Lama hujan	70
Debit sungai	12

**Nilai Output**

Prediksi	0.5
----------	-----

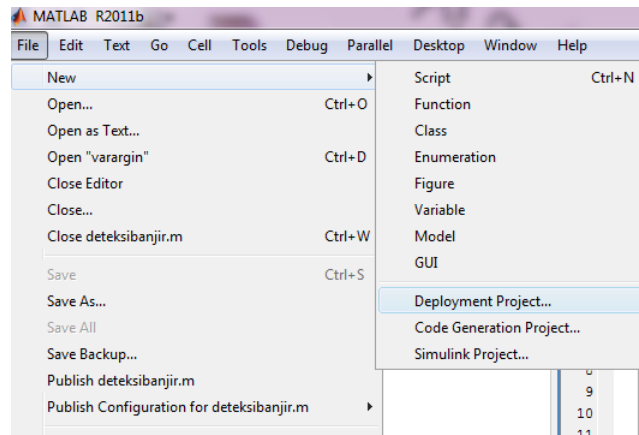
RUNNING

KELUAR

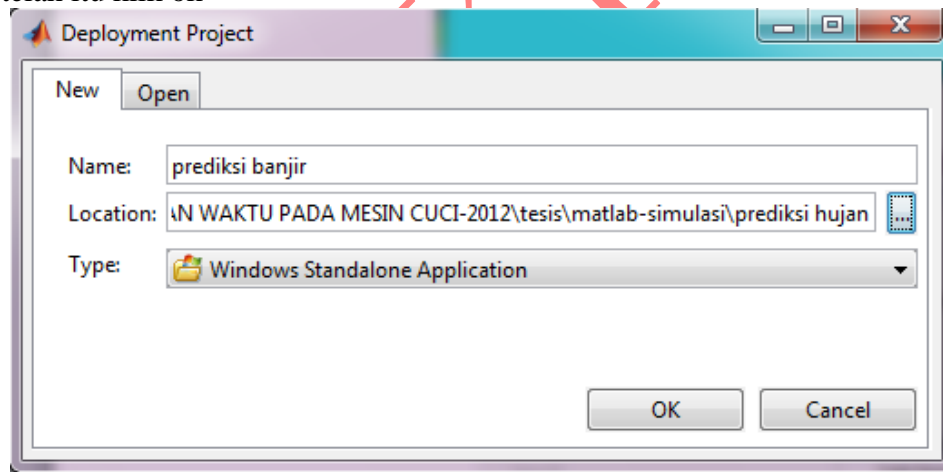
## Pembuatan program executable

Langkah penting selanjutnya adalah pembuatan program executable agar dapat diinstall dan dipakai di komputer lain tanpa harus memiliki ataupun membuka program matlab

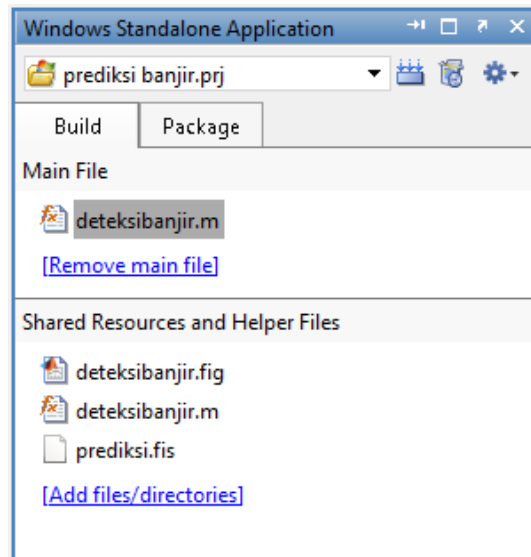
Agar proses berjalan cepat dan lancar, usahakan agar program yang terbuka pada komputer anda hanya program matlab, kemudian klik file/new/deployment project.



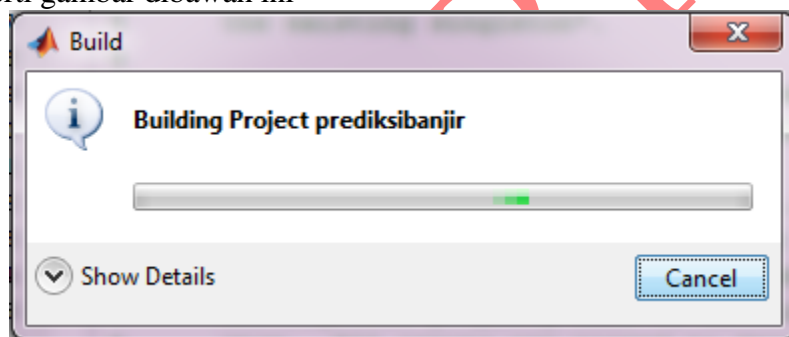
Pada jendela deployment, buatlah nama project, lokasi penyimpanan project serta tipe dari jenis aplikasi, setelah itu klik ok



Kemudian pada jendela standalone aplikasi anda diminta untuk memasukan file – file pendukung, serta memasukan MRC installer dan membuat Package. Lalu klik build



Jika semua file – file pendukung yang dibutuhkan telah terpilih semua maka proses build akan berlangsung seperti gambar dibawah ini



Setelah proses build selesai maka pada folder yang telah anda pilih sebagai tempat penyimpanan akan muncul atau tercipta file – file lain, dan itu berarti anda sudah berhasil membuat program aplikasi tersebut, dan bisa langsung mencobanya

Name	Date modified	Type	Size
prediksibanjir	9/23/2013 1:58 AM	File folder	
deteksibanjir	9/23/2013 1:25 AM	3dsMax biped figu...	6 KB
deteksibanjir.m	9/23/2013 1:25 AM	M File	8 KB
MCRInstaller	8/14/2011 6:58 PM	Application	406,320 KB
prediksi.fis	9/22/2013 12:19 PM	FIS File	2 KB
prediksibanjir	9/23/2013 2:01 AM	Application	282 KB
prediksibanjir.prj	9/23/2013 2:03 AM	PRJ File	52 KB
prediksibanjir_pkg	9/23/2013 2:03 AM	Application	406,273 KB
readme	9/23/2013 2:01 AM	Text Document	1 KB