

## I. GUI Matlab

GUI adalah singkatan dari Graphical User Interface, sebuah aplikasi display dari MATLAB yang mengandung tugas, perintah, atau komponen program yang mempermudah user (pengguna) dalam menjalankan sebuah program dalam MATLAB. GUI adalah interface yang dibangun dengan obyek grafik seperti tombol, kotak, panel, teks, slider, dan menu agar mudah di pelajari dan digunakan khalayak orang banyak. Tujuan dari penggunaan GUI yaitu membuat program terlihat lebih simpel dan praktis bagi para end-user. Oleh karena itu, hal yang perlu diperhatikan dalam membuat GUI adalah bagaimana menampilkan user interface yang mudah digunakan (user friendly) sekalipun user adalah orang awam yang tidak memahami kerumitan program di dalamnya.

## II. Analisis Regresi Sederhana

Analisis/uji regresi merupakan suatu kajian dari hubungan antara satu variabel, yaitu variabel yang diterangkan (the explained variabel) dengan satu atau lebih variabel, yaitu variabel yang menerangkan (the explanatory). Persamaan Regresi Linier Sederhana Persamaan regresi linier sederhana merupakan suatu model persamaan yang menggambarkan hubungan satu variabel bebas/ predictor (X) dengan satu variabel tak bebas/ response (Y). Persamaan regresi linier sederhana secara yaitu :

$$\hat{Y} = a + bX$$

dimana :  $\hat{Y}$  = garis regresi/ variable response

a = konstanta (intersep), perpotongan dengan sumbu vertikal

b = konstanta regresi (slope) X = variabel bebas/ predictor

## III. Studi Kasus

Studi kasus yang digunakan pada tugas ini adalah Angka Harapan Hidup (X1) dan Indeks Pembangunan Manusia (Y) tahun 2018. Maka buatlah prediksi Y untuk studi kasus di atas.

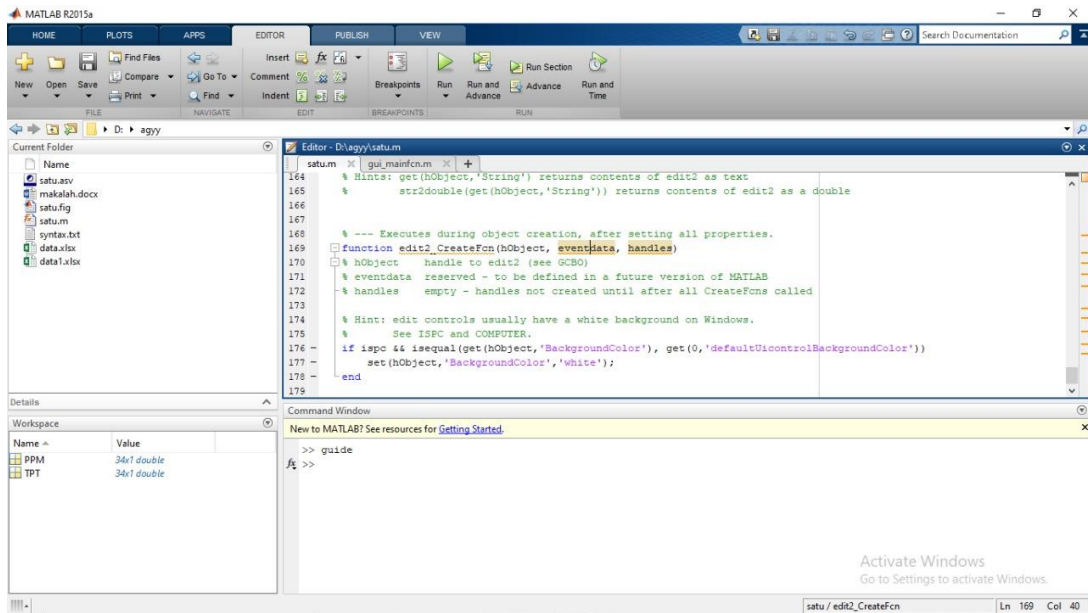
Kabupaten/Kota Se Jawa Timur	IPM	AHH
Kabupaten Pacitan	67,33	71,52
Kabupaten Ponorogo	69,91	72,43
Kabupaten Trenggalek	68,71	73,35
Kabupaten Tulungagung	71,99	73,74
Kabupaten Blitar	69,93	73,16
Kabupaten Kediri	71,07	72,37
Kabupaten Malang	69,4	72,26
Kabupaten Lumajang	64,83	69,7
Kabupaten Jember	65,96	68,74
Kabupaten Banyuwangi	70,06	70,34
Kabupaten Bondowoso	65,27	66,27

Kabupaten Situbondo	66,42	68,73
Kabupaten Probolinggo	64,85	66,71
Kabupaten Pasuruan	67,41	70,01
Kabupaten Sidoarjo	79,5	73,82
Kabupaten Mojokerto	72,64	72,24
Kabupaten Jombang	71,86	72,04
Kabupaten Nganjuk	71,23	71,25
Kabupaten Madiun	71,01	70,97
Kabupaten Magetan	72,91	72,3
Kabupaten Ngawi	69,91	71,92
Kabupaten Bojonegoro	67,85	71,07
Kabupaten Tuban	67,43	71,01
Kabupaten Lamongan	71,97	72,04
Kabupaten Gresik	75,28	72,46
Kabupaten Bangkalan	62,87	69,94
Kabupaten Sampang	61	67,79
Kabupaten Pamekasan	65,41	67,22
Kabupaten Sumenep	65,25	70,94
Kota Kediri	77,58	73,8
Kota Blitar	77,58	73,36
Kota Malang	80,89	72,93
Kota Probolinggo	72,53	70
Kota Pasuruan	74,78	71,18
Kota Mojokerto	77,14	73,01
Kota Madiun	80,33	72,59
Kota Surabaya	81,74	73,98
Kota Batu	75,04	72,37

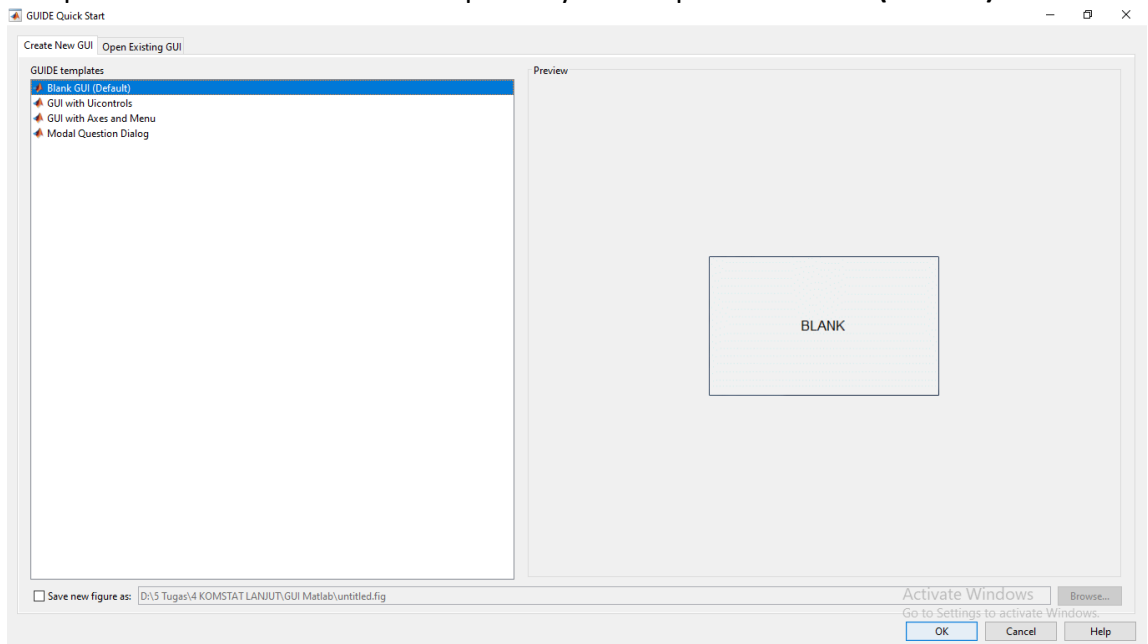
#### IV. Langkah Pembuatan GUI

##### MATLAB GUI Initialization

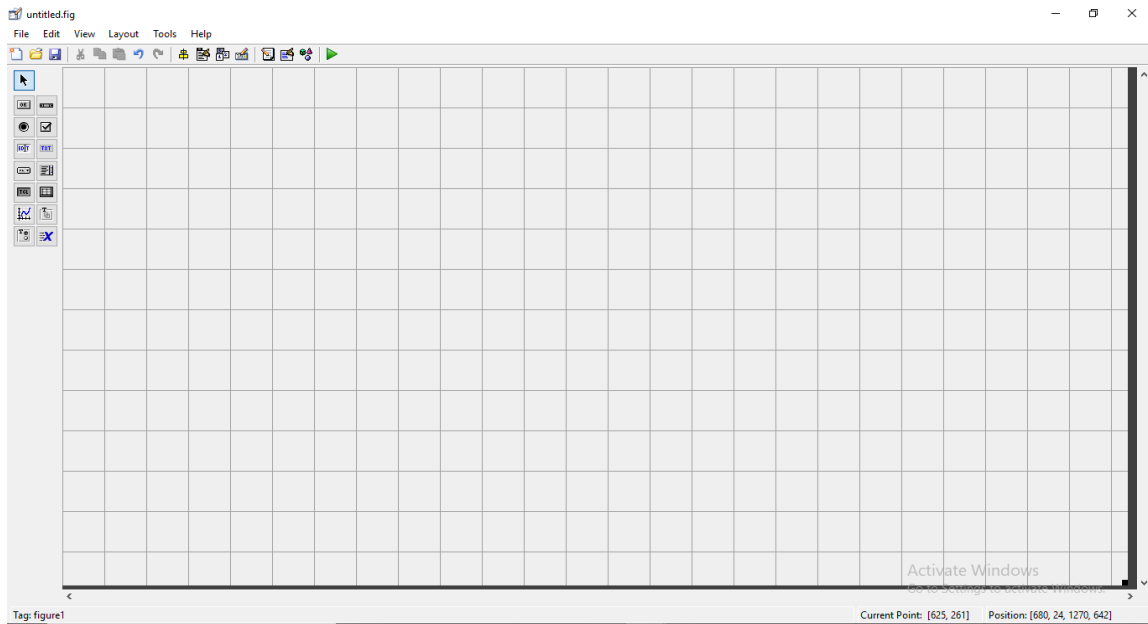
1. Buka MATLAB dan di command window ketik: guide



2. Tampilan dibawah ini akan muncul pada layar. Lalu pilih **Blank GUI (Default)** lalu klik **OK**.

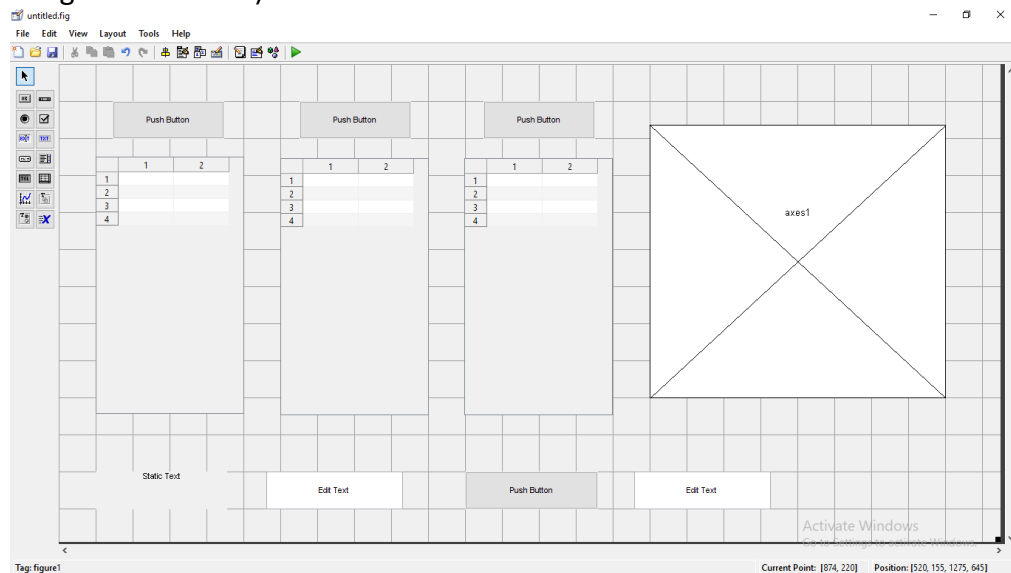


3. Tampilan sebuah GUI kosong akan muncul di layar.

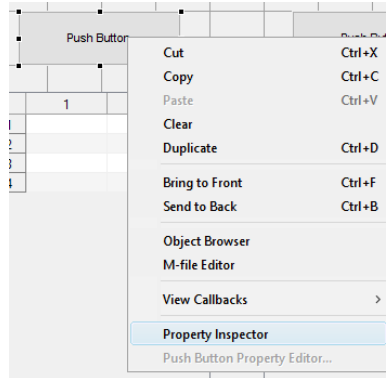


## Make a Regression Analyses GUI

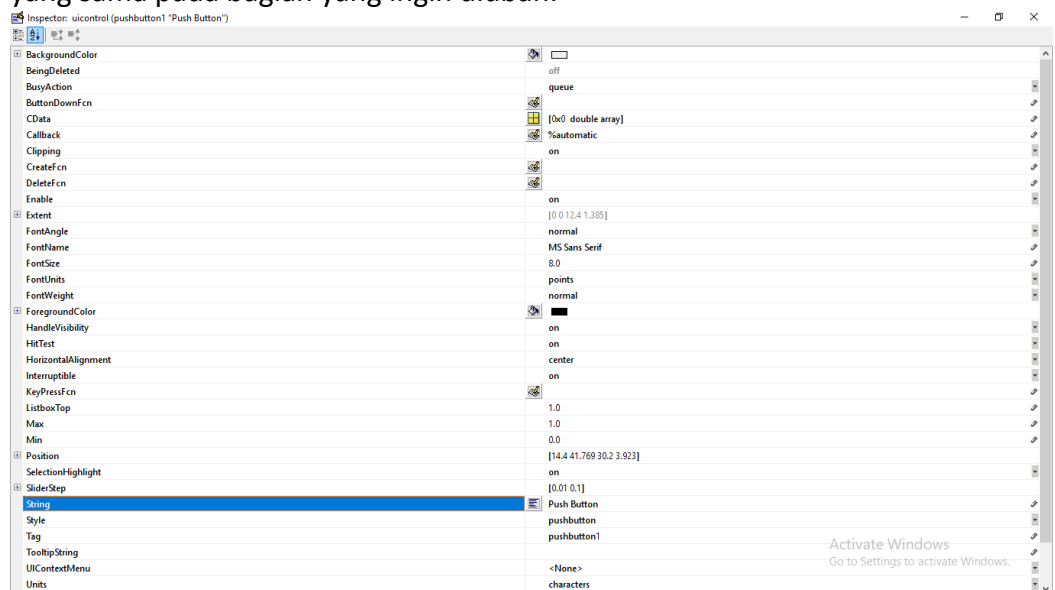
1. Buatlah tampilan GUI sesuai yang diinginkan (Gambar dibawah sebagai contoh) dengan menggunakan berbagai macam tools yang tersedia pada GUI Matlab yaitu 4 Push Button , 3 Table, 2 Static Text, 2 Edit Text, dan 1 axes (atau disesuaikan dengan kebutuhan).



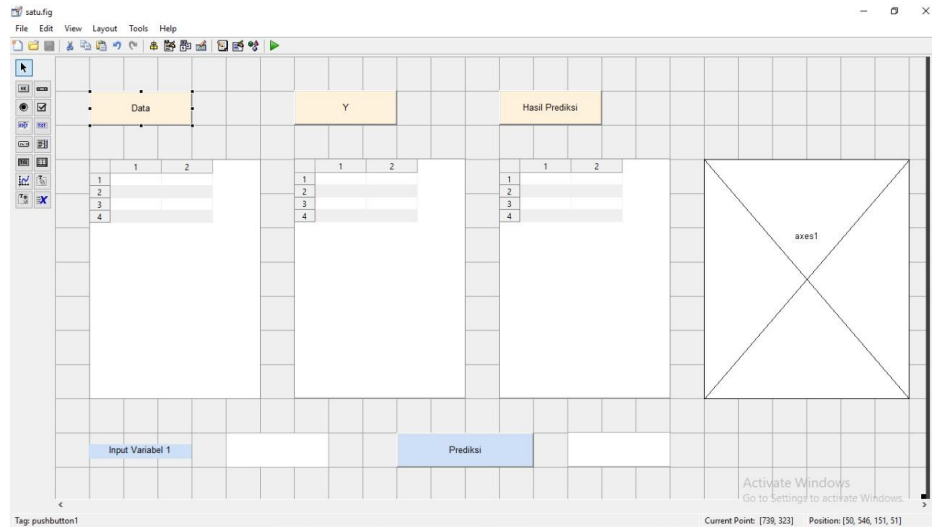
2. Untuk mengubah text yang terdapat pada push button, static test, dan edit text masuk pada pilihan Properties Inspector.
  - Klik Kanan pada bagian edit text/static test/push button -> kemudian pilih "Property Inspector".



3. Muncul window Property Inspector seperti tampilan di bawah ini, pada bagian “String” isi sesuai dengan text yang diinginkan, kemudian jika menginginkan bisa mengubah warna latar background pada bagian “Background Color”. Lakukan hal yang sama pada bagian yang ingin diubah.

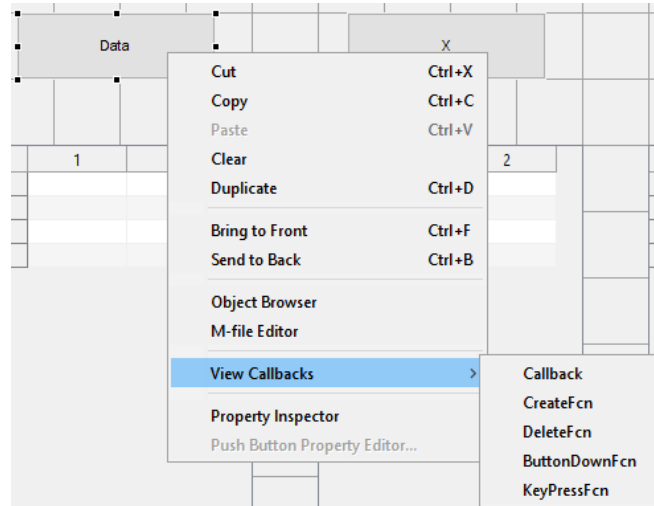


4. Setelah dilakukan perubahan pada Property Inspector, tampilan pada GUI akan seperti pada gambar dibawah.



5. Pada bagian push button Data.

- Klik Kanan -> View Callbacks -> Callback.



Sehingga terbuka gui\_regresi.m pada bagian pushbutton1\_callback.

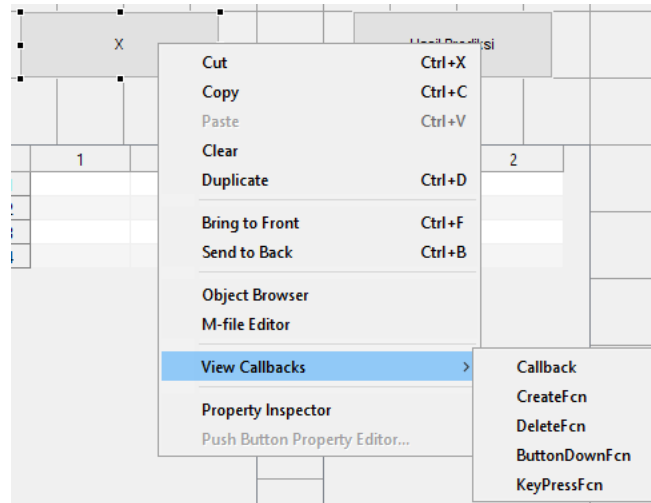
- Dibagian akhir dari pushbutton1\_callback tambahkan script seperti dibawah ini (script disesuaikan dengan nama studi kasus yang digunakan).

```
data=xlsread('data.xlsx','B2:C35');
handles.data=data;
set(handles.uitable1,'Data',data);
guidata(hObject,handles);
```

Note : Script ini untuk memanggil data yang sudah disiapkan dalam format excel (.xls/.xlsx)

6. Pada bagian push button X.

- Klik Kanan -> View Callbacks -> Callback.



Sehingga terbuka gui\_regresi.m pada bagian pushbutton2\_callback.

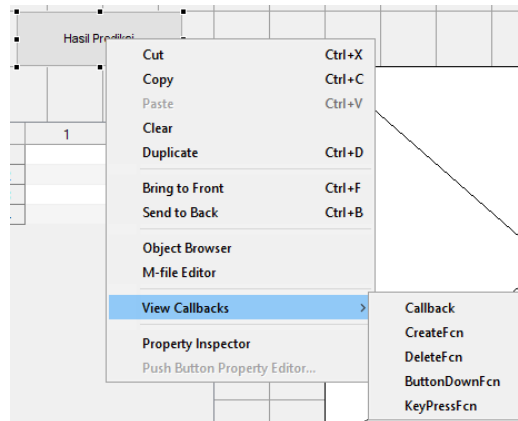
- Dibagian akhir dari pushbutton2\_callback tambahkan script seperti dibawah ini (script disesuaikan dengan nama studi kasus yang digunakan).

```
data=handles.data;
xtes=data(1:34,1);
handles.xtes=xtes;
set(handles.uitable2,'Data',xtes);
guidata(hObject,handles);
```

Note : Script ini untuk menampilkan data yang ingin ditampilkan.

#### 7. Pada bagian push button Hasil Produksi.

- Klik Kanan -> View Callbacks -> Callback.



Sehingga terbuka gui\_regresi.m pada bagian pushbutton3\_callback.

- Dibagian akhir dari pushbutton3\_callback tambahkan script seperti dibawah ini (script disesuaikan dengan nama studi kasus yang digunakan).

```

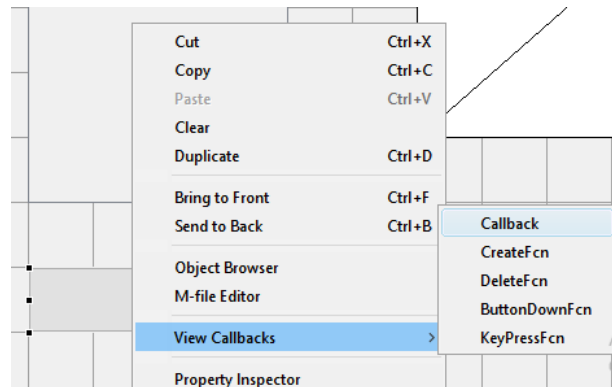
data=handles.data;
x=data(:,1);
y=data(:,2);
xtes=data(1:34,1);
ytes=data(1:34,2);
model=fitlm(x,y);
handles.model=model;
yp=predict(model,xtes);
axes(handles.axes1);
plot(x,y,'.g');
hold on
plot(xtes,ytes,'*b');
plot(xtes,yp,'-r');
hold off
set(handles.uitable3,'Data',yp);
guidata(hObject,handles);

```

Note : Script ini untuk menampilkan hasil prediksi data dan juga pemodelan analisis regresi sederhana dalam bentuk grafik plot.

8. Pada bagian push button Prediksi.

- Klik Kanan -> View Callbacks -> Callback.



Sehingga terbuka gui\_regresi.m pada bagian pushbutton4\_callback.

- Dibagian akhir dari pushbutton4\_callback tambahkan script seperti dibawah ini (script disesuaikan dengan nama studi kasus yang digunakan).

```

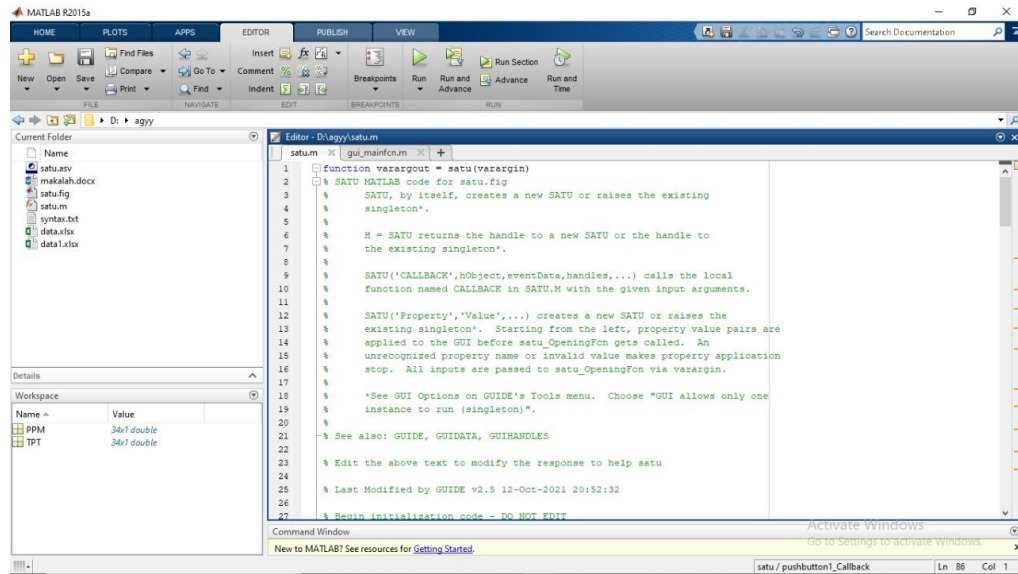
% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
data=xlsread('data.xlsx','B2:C39');
handles.data=data;
set(handles.uitable1,'Data',data);
guidata(hObject,handles);

```

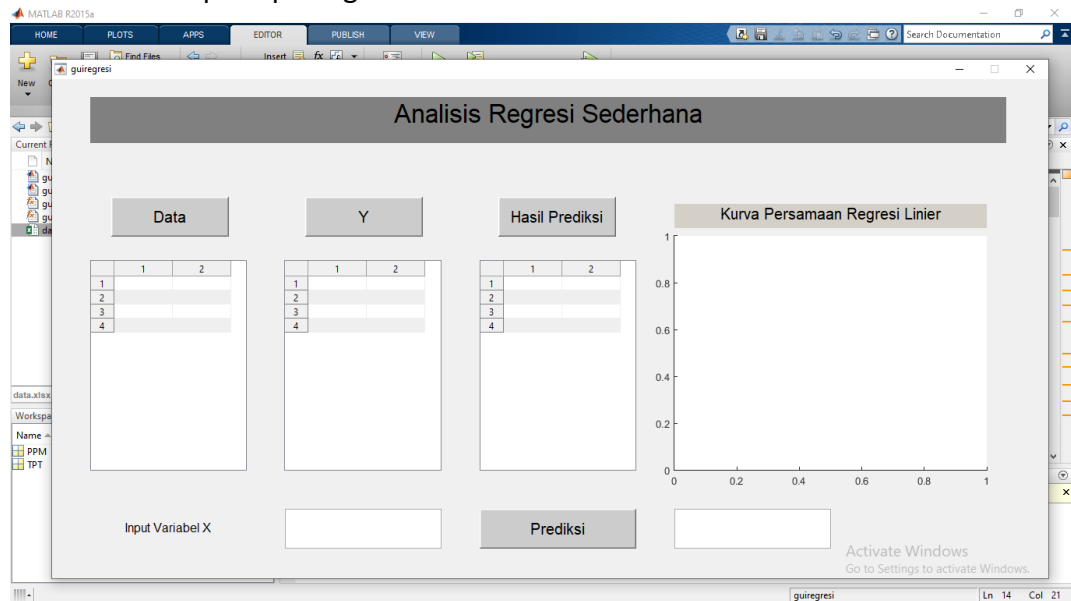
Note : Script ini untuk menampilkan hasil prediksi (y prediksi) pada masing masing data x.

9. Setelah dilakukan input syntax, pada bagian editor tekan "Run".



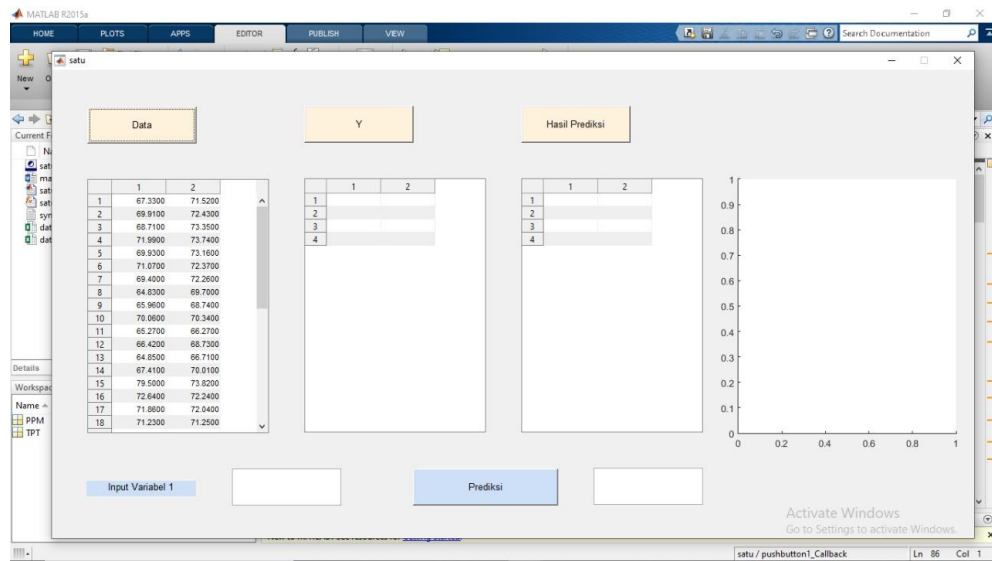


Lalu muncul seperti pada gambar dibawah.

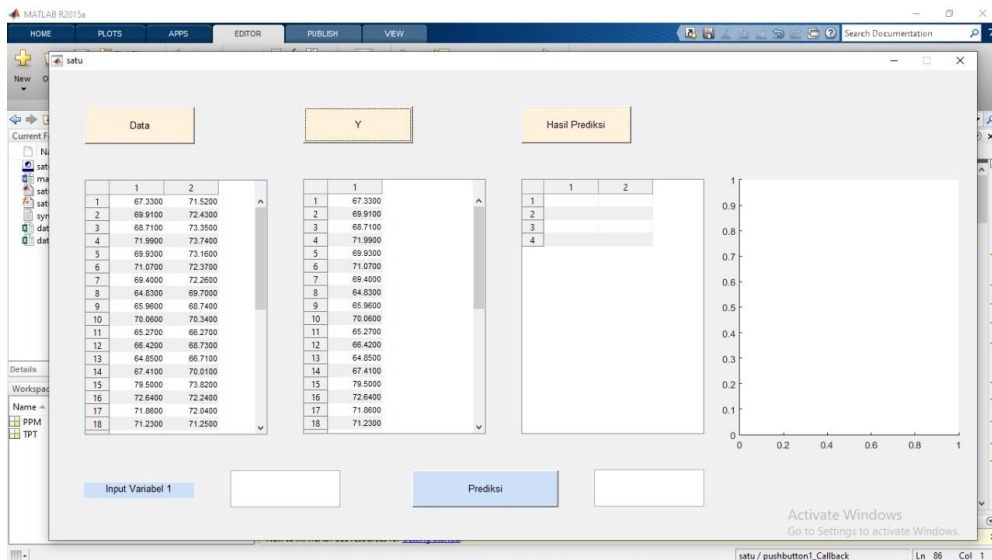


## V. Trial GUI Matlab

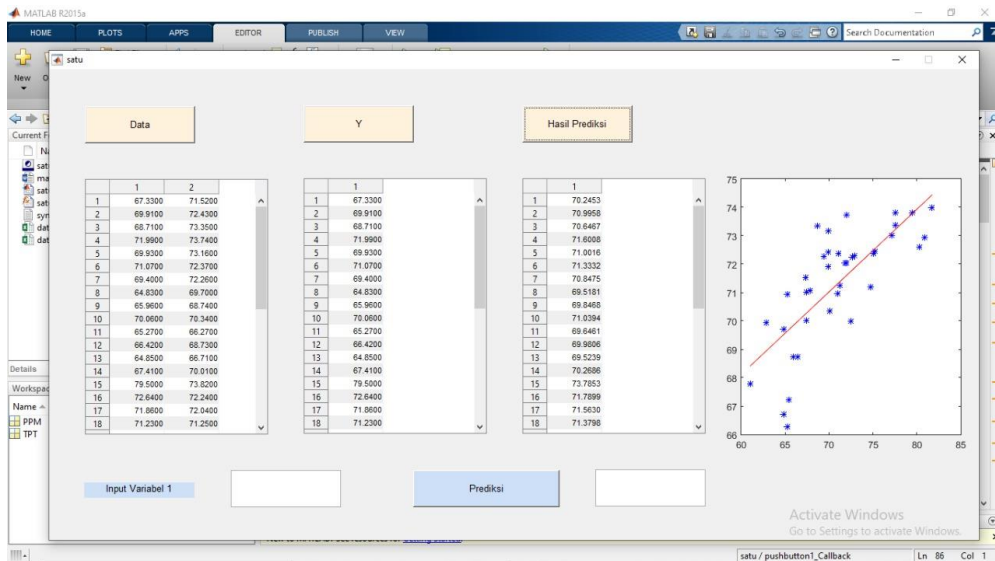
Menampilkan data.



Menampilkan variabel Y.



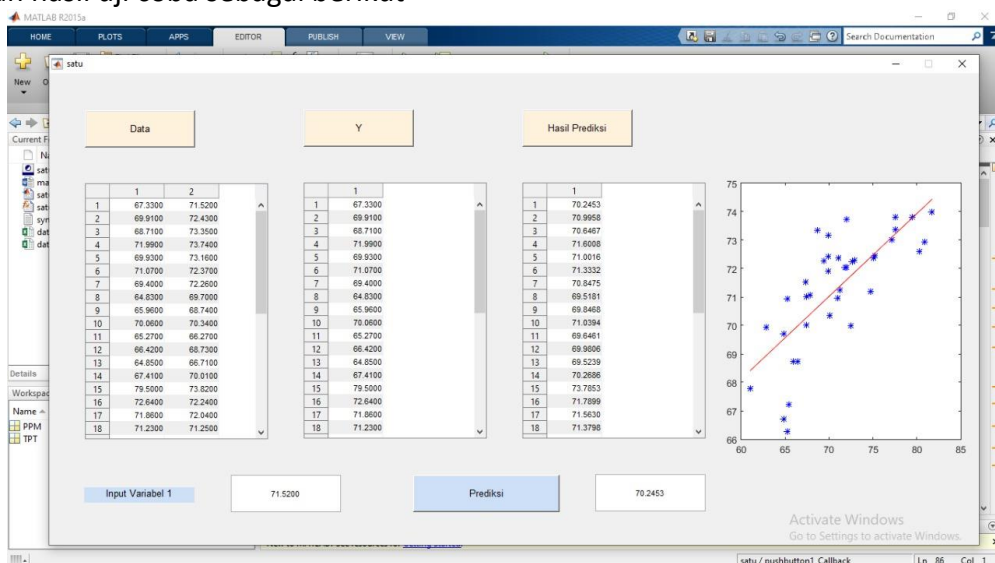
Menampilkan Hasil Prediksi, yang akan memunculkan Kurva Persamaan Regresi Linier.



Jika ingin mengecek salah satu hasil prediksi dari salah satu variabel X, dapat dicari melalui bagian dibawah ini.

Input Variabel X

Ditemukan hasil uji coba sebagai berikut



### Interpretasi:

Studi Kasus Pengaruh Angka Harapan Hidup Terhadap Indeks Pembangunan Manusia Menurut Kabupaten dan Kota di Jawa Timur Tahun 2018.

Berdasarkan hasil grafik yang keluar pada output dapat dilihat bahwa garis merah pada gambar adalah garis prediksi dan garis tersebut condong ke kanan yang berarti nilai Y dan X berbanding lurus, artinya semakin besar nilai X (Angka Harapan Hidup) maka semakin besar pula nilai Y (Indeks Pembangunan Manusia). Dan dari 38 data tersebut dengan menggunakan GUI Matlab untuk metode Analisis Regresi Linear Berganda, kita dapat memprediksi Indeks Pembangunan

Manusia di suatu daerah dengan memasukkan Angka Harapan Hidup pada daerah tersebut yang nantinya akan keluar nilai prediksi dari hasil pengolahan yang telah dibuat.

## LAMPIRAN

- Syntax Matlab Analisis Regresi Sederhana.

```
function varargout = guiregresi(varargin)
% GUIREGRESI MATLAB code for guiregresi.fig
%     GUIREGRESI, by itself, creates a new GUIREGRESI or raises the
existing
%     singleton*.
%
%     H = GUIREGRESI returns the handle to a new GUIREGRESI or the handle
to
%     the existing singleton*.
%
%     GUIREGRESI('CALLBACK',hObject,eventData,handles,...) calls the
local
%     function named CALLBACK in GUIREGRESI.M with the given input
arguments.
%
%     GUIREGRESI('Property','Value',...) creates a new GUIREGRESI or
raises the
%     existing singleton*. Starting from the left, property value pairs
are
%     applied to the GUI before guiregresi_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to guiregresi_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 guiregresi

% Last Modified by GUIDE v2.5 12-Oct-2021 19:05:24

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @guiregresi_OpeningFcn, ...
                  'gui_OutputFcn',  @guiregresi_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
```

```

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

% --- Executes just before guiregresi is made visible.
function guiregresi_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 guiregresi (see VARARGIN)

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

% Update handles structure
guidata(hObject, handles);

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

% --- Outputs from this function are returned to the command line.
function varargout = guiregresi_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 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('data.xlsx','B2:C39');
handles.data=data;
set(handles.uitable1,'Data',data);
guidata(hObject,handles);

% --- 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)

```

```

data=handles.data;
xtes=data(1:38,1);
handles.xtes=xtes;
set(handles.uitable2,'Data',xtes);
guidata(hObject,handles);

```

```

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton3 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
data=handles.data;
x=data(:,1);
y=data(:,2);
xtes=data(1:38,1);
ytes=data(1:38,2);
model=fitlm(x,y);
handles.model=model;
yp=predict(model,xtes);
axes(handles.axes1);
plot(x,y,'.g');
hold on
plot(xtes,ytes,'*b');
plot(xtes,yp,'-r');
hold off
set(handles.uitable3,'Data',yp);
guidata(hObject,handles);

```

```

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

```

```

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton4 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
model=handles.model;
x1=str2double(get(handles.edit1,'String'));
yt=predict(model,x1);
set(handles.edit2,'String',num2str(yt));
guidata(hObject,handles);

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

```